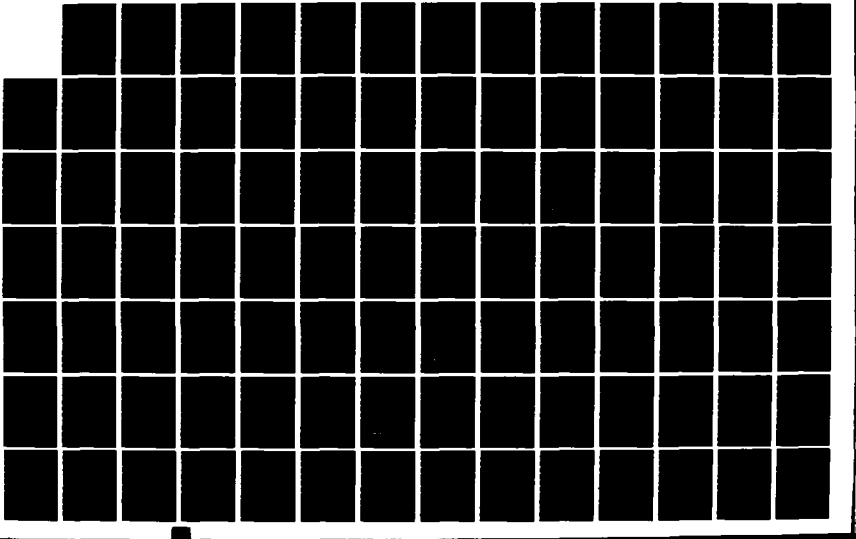
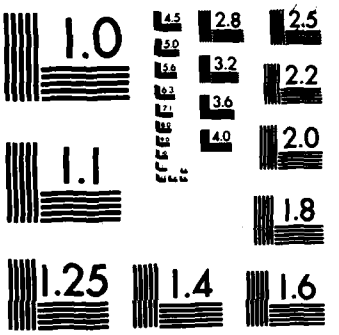
AD-A131 273 AFWL BENCHMARK COMPUTER PROGRAM(U) BOEING AEROSPACE CO 1/2
SEATTLE WA B J HENDERSON 01 MAY 82 DNA-5852F
DNA001-80-C-0056

UNCLASSIFIED F/G 9/2 NL





(12)

DNA 5852F

AFWL BENCHMARK COMPUTER PROGRAM

B. J. Henderson
Boeing Aerospace Company
P.O. Box 3999
Seattle, Washington 98124

1 May 1982

Final Report for Period 3 March 1980—1 May 1982

CONTRACT No. DNA 001-80-C-0056

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK WAS SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B310080464 P99DAXDB08853 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington,DC 20305

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: STTI, WASHINGTON, D.C. 20305, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH TO BE DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DNA 5852F	2. GOVT ACCESSION NO. A131273	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* AFWL BENCHMARK COMPUTER PROGRAM		5. TYPE OF REPORT & PERIOD COVERED Final Report for Period 3 March 1980 - 1 May 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) B. J. Henderson		8. CONTRACT OR GRANT NUMBER(s) DNA 001-80-C-0056
9. PERFORMING ORGANIZATION NAME AND ADDRESS Boeing Aerospace Company P.O. Box 3999 Seattle, Washington 98124		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask P99DAXDB088-53
11. CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D. C. 20305		12. REPORT DATE 1 May 1982
		13. NUMBER OF PAGES 96
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A since UNCLASSIFIED

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B310080464 P99DAXDB08853 H2590D.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Computer Benchmark
ANSI Fortran
Translator

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report includes the description of a computer program that partially translates IBM, CDC, DEC and VAX FORTRAN into the subset language of ANSI FORTRAN 77. This contract supports a DNA task to provide "fair" benchmark computer programs to vendors who bid on a DNA computer procurement project.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

PREFACE

This is the final report of an effort conducted under Defense Nuclear Agency Contract DNA-80-C-0056. The contract supported a DNA task of procuring an advanced scientific computer system by collecting representative sofware for evaluating the performance of candidate computer systems (benchmarking, Ref. 3). One sample benchmark program, a large hydrodynamics code named CSQ, has been processed into ANSI Standard Fortran using SCOFF, a translating computer program developed on this contract. The resulting version of CSQ has been tested and is now available in the DNA Cyber 176 computer located at Kirtland AFB, New Mexico.

This study was sponsored by the Office of Technical Information with LCDR Harold R. Gladwin serving as technical monitor. Major John M. Anderson reviewed the final report. Their comments and support are gratefully acknowledged. Special appreciation is extended to Michael Warshaw of RDA whose patience, cooperation, and advice provided great support for this report. The technical work of BAC staff members Loren Milliman and Dave Cruikshank made the successful completion of this contract possible.

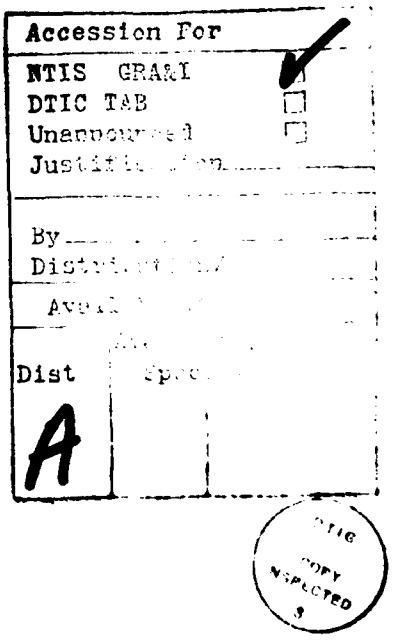


TABLE OF CONTENTS

LIST OF TABLES

SECTION 1

INTRODUCTION

Although the Fortran language was designed to be a universal and machine-independent medium for development of computer programs, computer system manufacturers have corrupted this by not fully implementing the ANSI Standard Language (Refs. 1,2) and by providing various enhancements to ANSI Fortran. Although a particular vendor's Fortran language may in itself be excellent, the irregularity makes it difficult to assemble a collection of Fortran programs which is free from non-standard coding. A computer program, SCOFF (Semi-automated Conversion of Fortran Four) was developed to translate programs written in the various dialects of the Fortran language to a standard, ANSI-compatible language. SCOFF converts most of the non-standard features of major vendor's Fortan to a standard Fortran language. For example, Control Data Corporation (CDC) Fortran allows variable names to be up to seven characters in length, while Digital Equipment Corporation (DEC) Fortran allows names up to 15 characters long which may contain dollar marks ($) and underscores(_). The ANSI Standard specified that variable names be a maximum of six characters in length and that all characters be from the set (A-Z, 0-9). SCOFF replaces each instance of an improper variable name with a truncated, proper variable name.

The SCOFF output meets the requirements of ANSI X3.9-1978 Fortran 77 Subset with two additions: that of COMPLEX statements and BLOCK Data structures. These additions were made because the preponderance of computer programs used in DNA activities require one or both of these features and because there is no convenient alternative.

SCOFF presently runs on IBM 360/370 series computers and requires approximately 200K bytes memory. SCOFF uses an efficient input/output method, so that the program execution time is comparable to the IBM-supplied copy utility IEBGENER. Several input formats (corresponding to source language maintenance program conventions) are accepted by SCOFF. Output may be in ASCII, EBCDIC, or BCD character codes. The Job Control Language required to execute the SCOFF program is shown in Table 4. A listing of SCOFF appears in Appendix B.

In Section 2 of this report, we describe the development of SCOFF. Section 3 briefly describes a demonstrated use of SCOFF.

SECTION 2

THE SCOFF COMPUTERIZED CONVERSION METHOD

SCOFF reads statements (lines or cards) one at a time from a Fortran program which is to be standardized. It then examines the symbols of the statements one at a time, comparing the symbols with a typical set such as a blank, $, *, C, or a numeral. This identification of the symbols determines the type of statement. The program then compares strings with standard statements, and takes appropriate action in translation.

A description of all the statements (ANSI and non-ANSI) that are recognized by SCOFF appears in Table 1, along with an indication of how SCOFF deals with the statement type. There are five actions that SCOFF may take, as specified in Table 2. Any statement type which does not appear in Table 1 will be ignored (passed through, not translated) by SCOFF. SCOFF cannot translate those statements which violate the ANSI Standards denoted by F in Table 1. There are several constructs for which there is no equivalent in the ANSI subset. The file manipulation statements OPEN and CLOSE are found in virtually all vendor Fortrans, but the format of parameters for OPEN or CLOSE varies considerably. Since neither OPEN nor CLOSE is defined in the ANSI Fortran Subset Language, SCOFF does not attempt to translate these statements. Also, SCOFF cannot resolve ambiguities which require syntactical analysis. The notation used in Table 1 is that of ANSI X3.9-1978, Fortran 77; a reprint of the conventions appears in Table 3.

Of the actions listed in Table 2, the first two are described below. The last three are self-explanatory.

2-1 RESTRUCTURING (R)

SCOFF does a global reorganization of program statements. Many vendor Fortrans allow Type Declaration statements and DATA statements to appear anywhere in a program element (or anywhere prior to the first occurrence of one of the defined variable names in an expression). SCOFF collects Type Declaration statements and DATA statements and places them ahead of the executible statements as required by the ANSI Fortran Subset. Also, some Fortrans allow data values to be specified in the Declaration statements. Any

Table 1. Statement handling by SCOFF

Initial Form	Action Taken	Translated Form
	(see Table 2)	
ASSIGN s TO i	T	ASSIGN s TO i
BACKSPACE u	T	BACKSPACE u
BACKSPACE (alist)	F	BACKSPACE (alist)
BLOCK DATA [sub]	N	BLOCK DATA [sub]
CALL sub [([a [,a]...])]	T	CALL sub [([a [,a]...])]
CHARACTER [*len[,]] nam [,nam]...	TI	CHARACTER [*len[,]] nam [,nam]...
CLOSE (cllist)	F	CLOSE (cllist)
COMMON [/[cb]/]nlist[[,]/[cb]/nlist]...	T	COMMON [/[cb]/]nlist[[,]/[cb]/nlist]...
COMPLEX v [,v]...	T	COMPLEX [,v]...
COMPLEX *len[,] v [,v]...	RT	COMPLEX v [,v]...
CONTINUE	N	CONTINUE
DATA nlist/clist/ [[,]nlist/clist]	T	DATA nlist/clist/ [[,]nlist/clist]
DIMENSION a(d) [,a(d)]...	T	DIMENSION a(d) [,a(d)]...
DO s [,] i=e_1,e_2 [,e_3]	T	DO s [,] i=e_1,e_2[,e_3]
DOUBLE PRECISION v[,v]...	RT	REAL v [,v]
ELSE	NI	ELSE
ELSE IF (e) THEN	TI	ELSE IF (e) THEN
END	N	END
END IF	NI	END IF
ENDFILE u	T	ENDFILE u
ENDFILE (alist)	F	ENDFILE (alist)
ENTRY en [([d [,d]...])]	TF	ENTRY en [([d [,d]...])]
EQUIVALENCE (nlist) [,(nlist]..	T	EQUIVALENCE (nlist) [,(nlist)]...
EXTERNAL proc [,proc]...	T	EXTERNAL proc [,proc]...

TABLE 1. (Continued)

Initial Form	Action Taken	Translated Form
FORMAT is	T	FORMAT is
fun ([d]...]) = e	T	fun ([d]...]) = e
[typ [*len]] FUNCTION fun([d [,d]...])	RT	[type] FUNCTION fun ([d [,d]...])
GO TO i [[,](s [,s]...)]	N	GO TO i [[,](s [,s]...)]
GO TO s	N	GO TO s
GO TO (s [,s]...)[,] i	T	GO TO (s [,s]...)[,] i
IF (e) st	T	IF (e) st
IF (e) s_1, s_2, s_3	T	IF (e) s_1, s_2, s_3
IF (e) s_1, s_2	F	IF (e) s_1, s_2
IF (e) THEN	TI	IF (e) THEN
IMPLICIT type (a [,a]...)	R	Specific Type Declaration Statements Generated As Required
INQUIRE (iflist) INQUIRE (iulist)	F F	INQUIRE (iflist) INQUIRE (iulist)
INTEGER v [,v]... INTEGER *len v [,v]...	T RT	INTEGER v [,v]... INTEGER v [,v]...
INTRINSIC fun [,fun]...	F	INTRINSIC fun [,fun]...
LOGICAL v [,v]... LOGICAL *len v[,v]...	T RT	LOGICAL v[,v]... LOGICAL v[,v]...
NAMELIST	TF	NAMELIST
OPEN (olist)	I	OPEN (olist)
PARAMETER (p=e[,p=e]...)	F	PARAMETER (p=e[,p=e]...)
PAUSE [n]	N	PAUSE [n]
PRINT f[, iolist]	T	PRINT f [, iolist]
PROGRAM pgm	T	PROGRAM pgm
READ (cilist) [iolist]	T	READ (cilist) [iolist]
READ f [, iolist]	F	READ f [, iolist]

TABLE 1. (Continued)

Initial Form	Action Taken	Translated Form
REAL v [,v]... REAL *len v [,v]	T RT	REAL v [,v]... REAL v [,v]...
RETURN [e]	N	RETURN [e]
REWIND u REWIND (alist)	T F	REWIND u REWIND (alist)
SAVE [a [,a]...]	F	SAVE [a [,a]...]
STOP [n]	N	STOP [n]
SUBROUTINE sub [([d [,d]...])]	RT	SUBROUTINE sub [([d [,d]...])]
WRITE (cilist) [iolist]	T	WRITE (cilist) [iolist]
v = e	T	v = e Arithmetic Assignment Statement
v = e	T	v = e Logical Assignment Statement
v = e	I	v = e Character Assignment Statement
v = v [=v...] = e	RT	v = e v = v [v = v] . . . Multiple Arithmetic Assignment Statement
st $ st [$st...]	RT	st st [st] . . . Multiple Statements per coding line
v(e(e[(e)...]))	RT	v = e(e) Subscripted Subscripts [v = e (v)] . . . v(v)
td v/clist/[,v|clist]...	RT	td v [,v]... Data in Type Declaration DATA v/clist/[,v/clist]

Table 2. Actions taken by SCOFF

Action	Description
R	Restructured by SCOFF to statement acceptable in ANSI subset
T	Variable names translated (truncated with ambiquity resolution) by SCOFF if longer than six characters
F	Flagged as not ANSI subset. Not translated by SCOFF unless TF appears
N	Translation not needed
I	Flagged as not acceptable to pre-1981 IBM compilers

Table 3. Notation of ANSI X3.9-1978 FORTRAN 77

In describing the form of FORTRAN statements or constructs, the following metalanguage conventions and symbols are used:

(1) Special characters from the FORTRAN character set, uppercase letters, and uppercase words are to be written as shown, except where otherwise noted.

(2) Lowercase letters and lowercase words indicate general entities for which specific entities must be substituted in actual statements. Once a given lowercase letter or word is used in a syntactic specification to represent an entity, all subsequent occurrences of that letter or word represent the same entity until that letter or word is used in a subsequent syntactic specification to represent a different entity.

(3) Brackets, [ ], are used to indicate optional items.

(4) An elipsis, ... , indicates that the preceding optional items may appear one or more times in succession.

(5) Blanks are used to improve readability, but unless otherwise noted have no significance.

(6) Words or groups of words that have special significance are underlined where their meaning is described.

An example illustrates the metalanguage. Given a description of the form of a statement as:

CALL sub[([a[, a]...])]

the following forms are allowed:

CALL sub
CALL sub ( )
CALL sub (a)
CALL sub (a, a)
CALL sub (a, a, a)
etc

When an actual statement is written, specific entities are substituted for sub and each a; for example:

CALL ABCD (X, 1.0)

Table 4. Job control language required to execute SCOFF

```
//  EXEC   PGM=SCOFF1, TIME=(0,20)
//STEPLIB   DD  DSN=ENG.BREL.PGMLIB,DISP=SHR
//OUTPUT    DD  SYSOUT=A,DCB=(BLKSIZE=400,RECFM=FB)
//TEMP      DD  UNIT=SYSDA,SPACE=(800, (200,20)), DCB=BLKSIZE=800
//FT06F001  DD  SYSOUT=A,DCB=BLKSIZE=1729
//INPUT     DD  *
```

The EXEC card specifies SCOFF1 to be run from library ENG.BREL. The STEPLIB statement defines where the program is in the library named ENG.BREL. The OUTPUT card image specifies the output file from SCOFF. The TEMP statement defines a temporary working file used for execution and storage of SCOFF. The FT06F001 statement defines FORTRAN unit six as the output unit. The INPUT statement tells the machine that the input program begins with the next statement.

such data values are removed from the Type Declaration statement and a separate DATA statement is created for the data value assignment.

Another global action taken by SCOFF is the replacement of IMPLICIT Type Statements. Any IMPLICIT statement is removed and individual Type Declaration statements are generated, if necessary, to explicitly type each variable name not already explicitly typed which does not correspond to the default type convention of ANSI Fortran (all names represent floating point variables unless the name begins with one of the letters I through N).

2-2 TRANSLATION (T)

Variable names that are more than six characters long are translated by truncation to make a similar, legal, unique name. Variable name truncation is done according to the following algorithm:

1. Drop all non-standard characters in the name.
2. If a vowel exists in the name, drop the last one; else drop the last consonant.
3. If the name is longer than six characters, do Step 2.
4. Set I to 6; set N to 0.
5. If the name is unique, then End.
6. Replace the Ith character with the character which succeeds it in the alphanumeric collating sequence.
7. Increment N. If N is less than 37, then do Step 5; else set I to I-1; set N to 0; do Step 6.

Another example of direct replacement is the Multiple Assignment Statement (permitted in CDC Fortran) which has the form

```
A = B = C = expression
```

A Multiple Assignment Statement is expanded into two (or more) single assignment Fortran statements:

```
C = expression
B = C
A = B
```

2-3 EXAMPLES

It may be helpful to examine two statements from Table 1 in more detail. Consider the first statement:

```
ASSIGN s TO i
```

The Action Taken is specified as T and the translated form appears to be identical with the initial form. The translated form will be identical except in those cases where the variable name i contains improper characters or is more than six characters long. SCOFF does not alter the statement segments ASSIGN, the statement number s, or TO.

Secondly, consider the statement:

```
COMPLEX*len[,] v [,v] ...
```

This statement is both Restructured and Translated (RT). The restructuring consists of removing the *len[,] while the translation ensures that every variable v is a legal name of six or fewer characters.

SECTION 3

DEMONSTRATED USES OF SCOFF

CSQ (Ref. 4) is a large two-dimensional hydrocode that was selected as a test vehicle for demonstrating the use of SCOFF. The selection was based upon the need for a large code representative of those used by the DNA community. CSQ can test the capabilities of computers proposed for purchase by DNA since it uses large amounts of both storage and CPU time.

CSQ is composed of approximately fifteen thousand cards. Most of the statements are pass-through statements with perhaps ten percent needing conversion. After application of SCOFF, a few days of hand work by a programmer were required to finish the conversion.

CSQ was first processed at Boeing via SCOFF. It was then transported to AFWL for trial runs on the DNA computer. The first runs were made under the NOS/BE system. Then when the operating system changed to NOS, some CSQ runs were made under the new operating system. The conversion was successful and made CSQ available as a benchmark code.

Appendix A presents a sample computer run of SCOFF. The input stream contains a number of statements that are non-ANSI. The output stream shows the translated statements. In addition, a set of SCOFF translation messages are presented. These messages would be used by a programmer if further translation was deemed necessary for the benchmark test.

SECTION 4

SUMMARY

SCOFF provides a positive step towards improving the development of fair, representative benchmarks for the evaluation of competing computer systems for DNA. The advantages of computer translation are apparent: the operation frees the programming staff from tedious work; it is fast, without error, and, within the limits described previously, complete. The changes which require a programmer's services are often challenging but usually such that they can be solved quickly.

Many programs already exist that are too large to translate by hand with any great confidence in the accuracy of the output. CSQ is an example of such a program. As larger computers and larger computer programs proliferate, the SCOFF technique will help reduce the cost of benchmarking and upgrading at a computer facility.

SECTION 5

REFERENCES

1. American National Standards Institute, "ANSI X3.9-1978 Fortran 77," New York (1978).

2. Lyon, G., Ed., "Using ANS FORTRAN," NBS Handbook 131, U.S. Department of Commerce/National Bureau of Standards, U.S. Gov. Printing Office, Washington (1980).

3. Conti. D. M., "Findings of the Standard Benchmark Library Study Group," Institute for Computer Science and Technology, U.S. Department of Commerce/National Bureau of Standards, U.S. Gov. Printing Office, Washington (1979).

4. Thompson, S. L., "CSQ--A Two-Dimensional Hydrodynamic Program With Energy Flow and Material Strength," Sandia Laboratories (SAND74-0122) Albuquerque, New Mexico (1975).

APPENDIX A
SAMPLE COMPUTER RUN OF SCOFF

This appendix presents a sample computer run of SCOFF that demonstrates typical inputs, outputs, and translation messages. The input program was written for illustrative purposes and does not solve any particular physics problem.

INPUT TO SCOFF

```
      PROGRAM TESTSCOFF
      PARAMETER (IDIM=100)
      IMPLICIT INTEGER*4 (R-S)
      DIMENSION ARRAYONE(IDIM),ARRAYTWO(IDIM)
      REAL*4 ARRAYTHREE(IDIM)/IDIM*0./
      INTEGER VARIABLE1,VARIABLE2/2/,VARIABLE3
      CHARACTER*1 ANS,YES
      EQUIVALENCE (ARRAYONE,ARRAYTHREE)
      DATA YES/'Y'/
      PRINT 5
    5 FORMAT(* THIS PROGRAM DOESNT DO ANYTHING!*)
      DO 500 VARIABLE1=1,IDIM
      VARIABLE3=(-1)**VARIABLE1
      ARRAYTWO(VARIABLE1)=ARRAYTHREE(VARIABLE1)+VARIABLE2*VARIABLE3
     &          *VARIABLE1
      IF(ARRAYTWO(VARIABLE1))100,100,500
  100 ARRAYTWO(VARIABLE1)=VARIABLE1*VARIABLE2
  500 CONTINUE
      PRINT 6
    6 FORMAT(* DO YOU WANT A PRINT?<Y OR N> *$)
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 7,(ARRAYTWO(R),R=1,IDIM)
    7 FORMAT(* ALL NUMBERS SHOULD BE POSITIVE COUNTING BY 2-->*,
     -(/1X,1P8E10.3))
      DO 600 VARIABLE2=1,IDIM
      ARRAYONE(VARIABLE2)=DIVIDEITUP(ARRAYTWO(VARIABLE2))
  600 CONTINUE
      PRINT 8
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 8,(ARRAYTHREE(S),S=1,IDIM)
    8 FORMAT(* ALL NUMBER SHOULD BE POSITIVE SEQUENTIAL-->*,
     -(/1X,1P8E10.3))
      CALL FLIPAROUND(ARRAYONE,IDIM,ARRAYTHREE)
      PRINT 6
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 9,ARRAYONE
    9 FORMAT(* ALL NUMBERS SHOULD BE SEQUENTIAL ALTERNATING SIGN-->*,
     -(/1X,1P8E10.3))
      PRINT 10
   10 FORMAT(* ALLLLLLDONE!!!*)
      STOP
      END
      REAL*4 FUNCTION DIVIDEITUP(RAY)
      DIVIDEITUP=RAY/2
      RETURN
      END
      SUBROUTINE FLIPAROUND(IIIIIII1,IIIIII,IIIIII2)
      REAL IIIIIII1(IIIIIII),IIIIII2(IIIIIII)
      DO 5000 J=1,IIIIII
      K=(-1)**J
      IIIIIII1(J)=IIIIII2(J)*K
 5000 CONTINUE
      RETURN
      END
```

OUTPUT FROM SCOFF

```
C=    PROGRAM TESTSCOFF
      PARAMETER (IDIM=100)
        INTEGER  R      , S                                          00000010
C=    IMPLICIT INTEGER*4 (R-S)
      DIMENSION ARRAYN(IDIM),ARRATW(IDIM)
      REAL ARRTHR(IDIM)
      INTEGER VARBL1,VARBL2,VARBL3
      CHARACTER*1 ANS,YES
      EQUIVALENCE (ARRAYN,ARRTHR)
      DATA   ARRTHR / IDIM*0. /
      DATA   VARBL2 / 2 /
      DATA YES/  1HY/
      PRINT 5
    5 FORMAT(33H THIS PROGRAM DOESNT DO ANYTHING!)
      DO 500 VARBL1=1,IDIM
      VARBL3=(-1)**VARBL1
      ARRATW(VARBL1)=ARRTHR(VARBL1)+VARBL2*VARBL3
     &          *VARBL1
      IF(ARRATW(VARBL1))100,100,500
  100 ARRATW(VARBL1)=VARBL1*VARBL2
  500 CONTINUE
      PRINT 6
    6 FORMAT(30H DO YOU WANT A PRINT?)Y OR N> S)
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 7,(ARRATW(R),R=1,IDIM)
    7 FORMAT(46H ALL NUMBERS SHOULD BE POSITIVE COUNTING BY 2-->,(/1X,1P
     -8E10.3))
      DO 600 VARBL2=1,IDIM
      ARRAYN(VARBL2)=DIVDTP(ARRATW(VARBL2))
  600 CONTINUE
      PRINT 6
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 8,(ARRTHR(S),S=1,IDIM)
    8 FORMAT(44H ALL NUMBER SHOULD BE POSITIVE SEQUENTIAL-->,(/1X,1P8E10
     -.3))
      CALL FLPRND(ARRAYN,IDIM,ARRTHR)
      PRINT 6
      READ *,ANS
      IF(ANS.EQ.YES)PRINT 9,ARRAYN
    9 FORMAT(53H ALL NUMBERS SHOULD BE SEQUENTIAL ALTERNATING SIGN-->,(/
     -1X,1P8E10.3))
      PRINT 10
   10 FORMAT(15H ALLLLLLDONE!!!)
      STOP
      END
      REAL*4 FUNCTION DIVDTP(RAY)
      DIVDTP=RAY/2
      RETURN
      END
      SUBROUTINE FLPRND(IIIIII,IIIIII,IIIII2)
      REAL IIIII1(IIIIII),IIIII2(IIIIII)
      DO 5000 J=1,IIIIII
      K=(-1)**J
      IIIIII(J)=IIII2(J)*K
 5000 CONTINUE
      RETURN
      END
```

```
                                        PROGRAM TESTSCOFF
  ---           PROGRAM TESTSCOFF                                                  ---
## SYNTAX ##
  ---           PARAMETER (IDIM=100)                                               ---
PARAMETER STATEMENT NOT ALLOWED IN ALL FORTRAN LANGUAGES
  ---           REAL*4 ARRAYTHREE(IDIM)/IDIM*0./                                   ---
                DATA   ARRAYTHREE / IDIM*0. /
  ---           INTEGER VARIABLE1,VARIABLE2/2/,VARIABLE3                           ---
                DATA   VARIABLE2 / 2 /
  ---           CHARACTER*1 ANS,YES                                                ---
CHARACTER STATEMENT NOT ALLOWED IN ALL FORTRAN LANGUAGES
  ---        6 FORMAT(* DO YOU WANT A PRINT?)Y OR N> *$)                           ---
CHARACTER ($) IS NOT A VALID FORMAT TYPE.
                INTEGER  R       S                                      00000010
 END OF TRANSLATION
```

APPENDIX B

LISTING OF SCOFF

```
          LOGICAL ENDCD
          LOGICAL OUPDAT, NUMLY
          REAL*8 ERRMSG(9), NAME(2)
          COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K
        1 RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM
        2  ,NERR,NERO,MEMKEY
          COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(13)
          COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE
          COMMON /DELTA/ MOVE, JCARD(80), ICARD(80), INBX(1520), IOL, NCARD
          COMMON /KAPPA/ KAP(11)
          COMMON /OMEGA/ KLAST(4),KSTOP(4)
          COMMON /SIGMA/ KSTIJ(11,60)
          COMMON / HASH / ILOC, LOC, NAME, IVAL, NUMLY
          COMMON /TAPES/ N1, N2, N3, N4, N5, N6
          COMMON /XUPDAT/ OUPDAT, ICDICT(27)
          DATA ERRMSG / 8HCHARCTR , 5HCLOSE, 4HELSE, 5HENDIF, 7HINQUIRE,
        +      8HINTRNSIC, 4HOPEN, 8HPARAMETR, 4HSAVE                    /
C
          IMPLGO = 0
          N1 = 8
          N2 = 11
          N3 = 12
          N4 = 6
          N5 = 5
          N6 = 6
          CALL OPEN
          MXCH = 7
        1 MINA = 37**2
          MAXA=27*MINA-1
          NUMLY = .FALSE.
          NROUT=0
          NRT=0
          NERR = 0
          NERO = 0
          MEMKEY = 0
          DO 2 I=1,8
2           KLEAR(I)=KBL
3         NEXT=MXLI
          NSYM=0
          NREC=0
          CALL HASHIN
C
C         IDENTIFY FORTRAN STATEMENT BY TYPE.
C
4         CALL INPUT
        5 KOL = 6
6         DO 8 J=1,69
            NIST=KSTIJ(11,J)
            LAST = KOL
            DO 7 I=1,NIST
 6001         LAST = LAST + 1
              IF (KARD(LAST).EQ.KSTIJ(I,J)) GO TO 7
              IF ( KARD(LAST) .NE. KBL ) GO TO 8
              GO TO 6001
7             CONTINUE
```

PAGE 1 OF MAIN

```
 7000     IF ( KARD(LAST+1) .NE. KBL ) GO TO 7001                                5600
          LAST = LAST + 1                                                        5700
          GO TO 7000                                                             5800
C                                                                                5900
C        GO TO FORTRAN STATEMENT PROCESSING BY TYPE.                             6000
C                                                                                6100
 7001     CONTINUE                                                               6200
          GO TO (90, 4,15,16,18,17,17,21,24,28,4,29,91,32,33,34,35,30,16,        6300
         1 40,41,42,43,44,99,45,4,18,46,46,49,49,54,54,54,54,56,64,65,92,        6400
         2 88,66,67,18,70,71,75,78,82,83,660,16,18,4,66,18,84,85,86,87,          6500
         3  6100,6200,6200,6200,6200,6600,6200,6800,6200                         6600
         4  ), J                                                                 6700
C                                                                                6800
8         CONTINUE                                                               6900
C                                                                                7000
C        NOT IN STATEMENT LIST.  ASSUME ARITHMETIC AND LOOK FOR =.               7100
C                                                                                7200
 9009    KOL = 6                                                                 7300
    9    LAST = 3                                                                7400
         ASSIGN 13 TO NOEQ                                                       7500
   10    K    = KMAX - 1                                                         7600
         DO 12 I=LAST,K                                                          7700
           IF ( KARD(I) .EQ. KSPK(1) ) GO TO 14                                  7800
   12    CONTINUE                                                                7900
         GO TO NOEQ, ( 4, 13, 205, 440, 460, 68 )                                8000
   13    CALL ERROR                                                              8100
         ITYPE=1                                                                 8200
         GO TO 11                                                                8300
   14    ITYPE = 2                                                               8400
         CALL MULTEQ( I )                                                        8500
         GO TO 4                                                                 8600
   11    CALL SUBSUB                                                             8700
         G O TO 4                                                                8800
C                                                                                8900
C              ************************************************                  9000
C              *  FORTRAN STATEMENT PROCESSING STARTS HERE  *                  9100
C              ************************************************                  9200
C                                                                                9300
C                                                                                9400
C        ASSIGN 12345 TO N.                                                      9500
C                                                                                9600
15       KOL=LAST                                                                9700
         CALL SYMBOL                                                             9800
         IF ( KODE .EQ. MINA ) CALL ERROR                                        9900
         ITYPE=1                                                                10000
         CALL UPDATE                                                            10100
         KOL=KOL+1                                                              10200
         CALL SYMBOL                                                            10300
         IF ( KODE .NE. MINA ) CALL ERROR                                       10400
         ITYPE=2                                                                10500
         CALL UPDATE                                                            10600
         GO TO 4                                                                10700
C                                                                               10800
C        BACKSPACE, ENDFILE, REWIND, ETC.                                       10900
C                                                                               11000
```

```
16      KOL=LAST                                                                11100
        ITYPE=1                                                                 11200
        CALL SYMBOL                                                             11300
        IF ( KODE .EQ. MINA ) CALL UPDATE                                       11400
        GO TO 4                                                                 11500
C                                                                               11600
C       BUFFER IN   ///   BUFFER OUT.                                           11700
C                                                                               11800
17      ITYPE=1                                                                 11900
        GO TO 25                                                                12000
C                                                                               12100
C       BLOCK DATA, FUNCTION, PROGRAM, SUBROUTINE.                              12200
C                                                                               12300
   18   ENDCD = .TRUE.                                                          12400
        IF (NREC.EQ.1.OR.NSYM.LT.2) GO TO 19                                    12500
        ENDCD = .FALSE.                                                         12600
        CALL ERR( 16,DUMMY,DUMMY )                                              12700
        IF ( OUPDAT ) CALL SORT( 1 )                                            12800
        NREC=1                                                                  12900
        NSYM=0                                                                  13000
        NEXT=MXLI                                                               13100
19      IF (J.NE.44.OR.NROUT.EQ.0) GO TO 20                                     13200
        NROUT=0                                                                 13300
        NRT=0                                                                   13400
   20   OUPDAT = .TRUE.                                                         13500
C---    ---- SET UP PAGE CAPTION FROM PREVIOUS CARD                             13600
        DO 201 I=1,LAST                                                         13700
  201     JOB(I) = KARD(I)                                                      13800
        DO 202 I=LAST,65                                                        13900
  202     JOB(I+1) = KARD(I)                                                    14000
C---    ---- REPLACE THE BLANK AFTER 'SUBROUTINE' OR WHATEVER                   14100
        JOB(LAST+1) = KBL                                                       14200
        IF ( KMAX .GT. 64 ) GO TO 204                                           14300
        DO 203 I=KMAX,64                                                        14400
  203     JOB(I+2) = KBL                                                        14500
  204   CALL CENTOR                                                             14600
  205   IF ( J .EQ. 5 ) GO TO 4                                                 14700
        KOL = LAST                                                              14800
        CALL FNCHEK( 0 )                                                        14900
        CALL RETRNI( 0 )                                                        15000
        GO TO 4                                                                 15100
C                                                                               15200
C       CALL XYZ (ARG LIST).                                                    15300
C                                                                               15400
21      KOL=LAST                                                                15500
        ITYPE=1                                                                 15600
        CALL SYMBOL                                                             15700
        IF ( KODE .NE. MINA ) CALL ERROR                                        15800
        ITYPE=4                                                                 15900
        IF ( KRSX .EQ. KSPK(1) ) GO TO 9009                                     16000
        CALL UPDATE                                                             16100
        GOTO 11                                                                 16200
C                                                                               16300
C       COMMON STATEMENT.                                                       16400
C                                                                               16500
```

```
   24   ITYPE = 5                                                                  16600
        KODE = 0                                                                   16700
        IF ( KARD(LAST+1) .NE. KSPK(4) ) GO TO 25                                  16800
C-------- ---- LABELLED COMMON.  SKIP LABEL                                       16900
        KOL = LAST+1                                                               17000
        CALL SYMBOL                                                                17100
        GO TO 26                                                                   17200
25      KOL=LAST                                                                   17300
26      CALL SYMBOL                                                                17400
        IF ( KODE ) 4,26,27                                                        17500
   27   CONTINUE                                                                   17600
        CALL UPDATE                                                                17700
        GO TO 26                                                                   17800
C                                                                                  17900
C       COMPLEX STATEMENT.                                                         18000
C                                                                                  18100
28      ITYPE=6                                                                    18200
        CALL TDTEST( &95, LAST, IMPLGO, 8, 16 )                                    18300
C-------- ---- TYPE DECLARATION STATEMENT.  SEE IF FOLLOWED BY "FUNCTION"        18400
  285   I1 = LAST                                                                  18500
        N = KSTIJ(11,28)                                                           18600
        DO 286 I=1,N                                                               18700
 2850     I1 = I1 + 1                                                            18800
          IF ( KARD(I1) .EQ. KSTIJ(I,28) ) GO TO 286                             18900
          IF ( KARD(I1) .NE. KBL ) GO TO 288                                     19000
          GO TO 2850                                                             19100
  286     CONTINUE                                                               19200
        LAST = I1                                                                  19300
        GO TO 18                                                                   19400
  288   CALL SLASHS( LAST )                                                        19500
        GO TO 4                                                                    19600
C                                                                                  19700
C       DATA STATEMENT.                                                            19800
C                                                                                  19900
29      ITYPE=7                                                                    20000
        ASSIGN 25 TO NOEQ                                                          20100
C-------- ---- LOOK FOR = OR /                                                    20200
  290   DO 292 I=LAST,KMAX                                                         20300
          IF ( KARD(I) .EQ. KSPK(1) ) GO TO 14                                   20400
          IF ( KARD(I) .EQ. KSPK(4) ) GO TO 294                                  20500
  292     CONTINUE                                                               20600
  294   GO TO NOEQ, ( 25, 830 )                                                    20700
C                                                                                  20800
C       DECODE, ENCODE.                                                            20900
C                                                                                  21000
30      KOL=LAST                                                                   21100
        ITYPE=1                                                                    21200
        CALL SYMBOL                                                                21300
        IF ( KODE .LT. 0 ) GO TO 4                                                 21400
31      CALL SYMBOL                                                                21500
        CALL UPDATE                                                                21600
        GO TO 26                                                                   21700
C                                                                                  21800
C       DIMENSION STATEMENT.                                                       21900
C                                                                                  22000
```

```
32      ITYPE=8                                                          22100
        GO TO 25                                                         22200
C                                                                        22300
C       DOUBLE PRECISION STATEMENT.                                      22400
C                                                                        22500
   33   KOL = 6                                                          22600
        CALL MOVER( 4,4HREAL,1,15 )                                      22700
        LAST = KOL                                                       22800
        KMAX = KMAX - KOL + 10                                           22900
        ITYPE = 18                                                       23000
        GO TO 265                                                        23100
C                                                                        23200
C       DOUBLE STATEMENT.                                                23300
C                                                                        23400
34      ITYPE=9                                                          23500
        GO TO 25                                                         23600
C                                                                        23700
C       DO STATEMENT.                                                    23800
C                                                                        23900
35      DO 36 I=LAST,KMAX                                                24000
          IF (KARD(I).EQ.KSPK(1)) GO TO 37                               24100
          IF (KARD(I).EQ.KSPK(3)) GO TO 9                                24200
36        CONTINUE                                                       24300
        CALL ERROR                                                       24400
        GO TO 4                                                          24500
37      IB=I                                                             24600
        DO 38 I=IB,KMAX                                                  24700
          IF (KARD(I).EQ.KSPK(2)) GO TO 39                               24800
38        CONTINUE                                                       24900
        GO TO 14                                                         25000
39      KOL=LAST                                                         25100
        ITYPE=20                                                         25200
        IF (KARD(I).EQ.KSPK(3)) GO TO 11                                 25300
        CALL SYMBOL                                                      25400
C                                                                        25500
        CALL UPDATE                                                      25600
        KOL=KOL-1                                                        25700
        ITYPE = 21                                                       25800
        GO TO 26                                                         25900
C                                                                        26000
C               ******************************************               26100
C               *              END STATEMENT             *               26200
C               *       EXECUTE SORT AND PRINT INDEX     *               26300
C               ******************************************               26400
C                                                                        26500
40      IF (KMAX.GT.LAST) GO TO 9                                        26600
        CALL SORT( 1 )                                                   26700
        GO TO 3                                                          26800
C                                                                        26900
C       ENTRY STATEMENT.                                                 27000
C                                                                        27100
   41   ASSIGN 205 TO NOEQ                                               27200
        GO TO 10                                                         27300
C                                                                        27400
C       EQUIVALENCE STATEMENT.                                           27500
```

```
C                                                                                27600
42      ITYPE=10                                                                 27700
        KOL=LAST+1                                                               27800
        GO TO 26                                                                 27900
C                                                                                28000
C       EXTERNAL STATEMENT.                                                      28100
C                                                                                28200
43      ITYPE=11                                                                 28300
        KOL = LAST                                                               28400
        CALL XTRNAL                                                              28500
        GO TO 4                                                                  28600
C                                                                                28700
C-------- ---- FIND STATEMENT                                                   28800
    44  ASSIGN 440 TO NOEQ                                                       28900
        GO TO 10                                                                 29000
   440  CALL ERROR                                                               29100
        GO TO 96                                                                 29200
C                                                                                29300
C       FORTRAN OR TYPE STATEMENT.                                               29400
C                                                                                29500
45      KOL=LAST                                                                 29600
        GO TO 6                                                                  29700
C                                                                                29800
C       GO TO (1,2,3,4,5), N   ///   GO TO 1   ///   GO TO N   ///   ETC.      29900
C                                                                                30000
    46  ASSIGN 460 TO NOEQ                                                       30100
        GO TO 10                                                                 30200
460     KOL=LAST                                                                 30300
        ITYPE=1                                                                  30400
47      CALL SYMBOL                                                              30500
        IF ( KODE ) 4,47,48                                                      30600
48      CALL UPDATE                                                              30700
        GO TO 47                                                                 30800
C                                                                                30900
C       IF ACCUMULATOR OVERFLOW STATEMENT.                                       31000
C                                                                                31100
49      DO 50 I=LAST,KMAX                                                        31200
          IF (KARD(I).EQ.KABC(23)) GO TO 51                                      31300
50        CONTINUE                                                               31400
        CALL ERROR                                                               31500
        GO TO 4                                                                  31600
51      KOL=I                                                                    31700
        ITYPE=1                                                                  31800
52      CALL SYMBOL                                                              31900
        IF ( KODE ) 4,52,53                                                      32000
53      IF (KODE.GE.MINA.AND.KODE.LE.MAXA) CALL ERROR                            32100
        CALL UPDATE                                                              32200
        GO TO 52                                                                 32300
C                                                                                32400
C       IF (DIVIDE CHECK) /// IF (END FILE) /// IF (SENSE LIGHT).                32500
C       IF (SENSE SWITCH) 1,2                                                    32600
C                                                                                32700
54      DO 55 I=LAST,KMAX                                                        32800
          IF (KARD(I).EQ.KSPK(5)) GO TO 51                                       32900
55        CONTINUE                                                               33000
```

```
      CALL ERROR                                                        33100
      GO TO 4                                                           33200
C                                                                       33300
C     IF (ARITH) 1,2,3   ///   IF (LOGICAL) STATEMENT.                  33400
C                                                                       33500
C     LOOK FOR PARENTHESIS COUNT OF ZERO.                               33600
C                                                                       33700
56    NPAR=0                                                            33800
      DO 58 I=LAST,KMAX                                                 33900
        IF (KARD(I).NE.KSPK(3)) GO TO 57                                34000
        NPAR=NPAR+1                                                     34100
        GO TO 58                                                        34200
57      IF (KARD(I).NE.KSPK(5)) GO TO 58                                34300
        NPAR=NPAR-1                                                     34400
        IF (NPAR) 59,59,58                                              34500
58      CONTINUE                                                        34600
      CALL ERROR                                                        34700
      GO TO 4                                                           34800
C                                                                       34900
C     SCAN SYMBOLS WITHIN PARENTHESIS.                                  35000
C                                                                       35100
59    J=KMAX                                                            35200
      IB=I                                                              35300
      KMAX=IB                                                           35400
      ITYPE=1                                                           35500
      KOL=LAST                                                          35600
      CALL SUBSUB                                                       35700
62    KOL=IB                                                            35800
      KMAX=J                                                            35900
C                                                                       36000
C     LOOK FOR DIGIT IMMEDIATELY AFTER PARENTHESIS.  IF ONE IS NOT      36100
C     FOUND, THE IF STATEMENT IS PROBABLY LOGICAL.                      36200
C                                                                       36300
      DO 63 I=1,10                                                      36400
        IF (KARD(KOL+1).EQ.KDIG(I)) GO TO 52                            36500
63      CONTINUE                                                        36600
      IF ( KARD(KOL+1) .NE. KBL ) GO TO 6                               36700
      IB = IB + 1                                                       36800
      GO TO 62                                                          36900
C                                                                       37000
C     INTEGER STATEMENT.                                                37100
C                                                                       37200
64    ITYPE=12                                                          37300
      CALL TDTEST( &95, LAST, IMPLGO, 2, 4 )                            37400
      GO TO 285                                                         37500
C                                                                       37600
C     LOGICAL STATEMENT.                                                37700
C                                                                       37800
65    ITYPE=13                                                          37900
      CALL TDTEST( &95, LAST, IMPLGO, 1, 4 )                            38000
      GO TO 285                                                         38100
C                                                                       38200
C     PAUSE, STOP, RETURN STATEMENT.                                    38300
C                                                                       38400
   66 ASSIGN 4 TO NOEQ                                                  38500
```

```
        !GO TO 10
C----!-!---- RETURN.
  660!  !IF ( LAST .GE. KMAX ) GO TO 4
C>>     !CALL RETRNI( 1 )
        !GO TO 4
C       !
C       !PRINT STATEMENT.
C       !
67      !ITYPE=15
68      !KOL=LAST
        !CALL SYMBOL
        !IF ( KODE .LT. 0 ) GO TO 4
C----!-!---- LOOK FOR COMMA BEFORE =
   69!  !IF ( KRSX .EQ. KSPK(2) ) GO TO 11
        !IF ( KRSX .EQ. KSPK(1) ) GO TO 14
        !IF ( KRSX .EQ. KSPK(3) ) GO TO 9
        !GO TO 4
C       !
C       !PUNCH STATEMENT.
C       !
70      !ITYPE=16
        !GO TO 68
C       !
C       !READ INPUT TAPE 5, 6, LIST.
C       !
71      !ITYPE=17
        !KOL=LAST+3
72      !CALL SYMBOL
        !IF ( KODE .LT. 0 ) GO TO 4
73      !IF (KODE.GE.MINA.AND.KODE.LE.MAXA) CALL UPDATE
        !CALL SYMBOL
        !IF (KODE) 4,4,74
74      !CALL UPDATE
        !GO TO 26
C       !
C       !READ TAPE 5, LIST
C       !
75      !ITYPE=17
76      !KOL=LAST
        !CALL SYMBOL
        !IF ( KODE .LT. 0 ) GO TO 4
77      !IF (KODE.GE.MINA.AND.KODE.LE.MAXA) CALL UPDATE
        !GO TO 26
C       !
C       !READ (5) LIST   ///   READ (5,6) LIST
C       !
78      !ITYPE=17
79      !KOL=LAST
        !CALL SYMBOL
        !IF ( KODE .LT. 0 ) GO TO 4
80      !IF (KODE.GE.MINA.AND.KODE.LE.MAXA) CALL UPDATE
        !IF (KRSX.EQ.KSPK(5)) GO TO 11
        !CALL SYMBOL
        !IF ( KODE .LT. 0 ) GO TO 4
   81!  !IF ( KRSX .EQ. KSPK(11) ; CALL ERROR
```

PAGE 8 OF MAIN

```
      IF ( KRSX .EQ. KSPK(1) ) CALL ERREND                                  44100
      GO TO 80                                                              44200
C                                                                           44300
C     READ 5, LIST                                                          44400
C                                                                           44500
82    ITYPE=17                                                              44600
      GO TO 88                                                              44700
C                                                                           44800
C     REAL STATEMENT.                                                       44900
C                                                                           45000
   83 ASSIGN 830 TO NOEQ                                                    45100
      GO TO 290                                                             45200
  830 ITYPE=18                                                              45300
      CALL TDTEST( &95, LAST, IMPLGO, 4, 8 )                                45400
      GO TO 285                                                             45500
C                                                                           45600
C     TYPE STATEMENT.                                                       45700
C                                                                           45800
84    KOL=LAST                                                              45900
      GO TO 6                                                               46000
C                                                                           46100
C     WRITE OUTPUT TAPE 5, 6, LIST                                          46200
C                                                                           46300
85    ITYPE=19                                                              46400
      KOL=LAST+5                                                            46500
      GO TO 72                                                              46600
C                                                                           46700
C     WRITE TAPE 5, LIST.                                                   46800
C                                                                           46900
86    ITYPE=19                                                              47000
      GO TO 76                                                              47100
C                                                                           47200
C     WRITE 5, LIST.                                                        47300
C                                                                           47400
87    ITYPE=19                                                              47500
      GO TO 79                                                              47600
C                                                                           47700
C     NAMELIST STATEMENT.                                                   47800
C                                                                           47900
88    ITYPE=14                                                              48000
      GO TO 25                                                              48100
C-----------  ---- ** MEMBER = NAME STATEMENT ( FROM UPDATER TAPE )         48200
   90 MOVE = -1                                                             48300
      NREC = NREC-1                                                         48400
      GO TO 4                                                               48500
C-----------  ---- DEFINE FILE STATEMENT                                    48600
   91 KOL = LAST                                                            48700
      GO TO 97                                                              48800
C                                                                           48900
C-----------  ---- IMPLICIT STATEMENT. TEST TO SEE IF FIRST STATEMENT IN PROGRAM 49000
C-----------  ---- PROCESS IMPLICIT STATEMENT                               49100
   92 IMPLGO = 1                                                            49200
      KOL = LAST                                                            49300
      GO TO 6                                                               49400
   95 KOL = LAST+2                                                          49500
```

```
                CALL IMPSET( IDX )                                                  49600
                IF ( IDX .EQ. 1 ) GO TO 6                                           49700
                IMPLGO = 0                                                          49800
     98         KARD(1) = KABC(3)                                                   49900
                KARD(2) = KSPK(1)                                                   50000
     97         CALL ERR( 19,0,0 )                                                  50100
                GO TO 4                                                             50200
C----+----      ---- FORMAT STATEMENT                                               50300
     99         CALL FORTRN( KMAX, KARD, .TRUE. )                                   50400
                MOVE = 2                                                            50500
                NCARD = NCARD-1                                                     50600
                GO TO 4                                                             50700
C                                                                                   50800
C----+----      ---- CHARACTER STATEMENT                                            50900
 6100           ITYPE = 3                                                          51000
                CALL ERR( 18,5,10H CHARACTER )                                      51100
                IF ( KARD(LAST+1) .NE. KSPK(8) ) GO TO 6120                         51200
                KOL = LAST + 2                                                      51300
                NUMLY = .TRUE.                                                      51400
                CALL SYMBOL                                                         51500
                LAST = KOL                                                          51600
                NUMLY = .FALSE.                                                     51700
 6120           IF ( IMPLGO  .EQ. 1 ) GO TO 95                                      51800
                GO TO 285                                                           51900
C----+----      ---- CLOSE                                                         52000
 6200           ITYPE = J - 40                                                     52100
                ASSIGN 6250 TO NOEQ                                                 52200
                GO TO 10                                                            52300
 6250           CALL ERR( 18,4,ERRMSG(ITYPE-20) )                                   52400
                GO TO 4                                                             52500
C----+----      ---- INTRINSIC FUNCTION                                            52600
 6600           CONTINUE                                                           52700
                CALL ERR( 18,5,10HINTRINSIC )                                       52800
                GO TO 4                                                             52900
 6800           CONTINUE                                                           53000
                CALL ERR( 18,5,10HPARAMETER )                                       53100
                GO TO 4                                                             53200
                  END                                                               53300
```

```
      BLOCK DATA                                                                100
C                                                                               200
C     THIS BLOCK DATA CONTAINS ALL THE INDEX DATA STATEMENTS.                   300
C                                                                               400
      LOGICAL*1 OTAB(256)                                                       500
      COMMON /TAPES/ N1, N2, N3, N4, N5, N6                                     600
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(13)                              700
      COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE,KSBO,KCDO,KOUT(80)                800
      COMMON /GAMMA / OTAB                                                      900
      COMMON /DELTA/ MOVE,JCARD(80), ICARD(1600), IOL, NCARD                   1000
     1   , OSTMTN, FRSTSB, IC, JC, IPAR, L1(25), KLASS(25)                     1100
      COMMON /KAPPA/ KAP(11)                                                   1200
      COMMON /OMEGA/ KLAST(4),KSTOP(4)                                         1300
      COMMON /SIGMA/ KST1(11),KST2(11),KST3(11),KST4(11),KST5(11),KST6(1       1400
     1)),KST7(11),KST8(11),KST9(11),KST10(11),KST11(11),KST12(11),KST13(      1500
     2)),KST14(11),KST15(11),KST16(11),KST17(11),KST18(11),KST19(11),KS       1600
     3)T20(11),KST21(11),KST22(11),KST23(11),KST24(11),KST25(11),KST26(11     1700
     4)),KST27(11),KST28(11),KST29(11),KST30(11),KST31(11),KST32(11),KST3     1800
     5)3(11),KST34(11),KST35(11),KST36(11),KST37(11),KST38(11),KST39(11),     1900
     6)KST40(11),KST41(11),KST42(11),KST43(11),KST44(11),KST45(11),KST46(     2000
     7)11),KST47(11),KST48(11),KST49(11),KST50(11),KST51(11),KST52(11),KS     2100
     8)T53(11),KST54(11),KST55(11),KST56(11),KST57(11),KST58(11),KST59(11     2200
     9)),KST60(11)                                                            2300
     A), KST61(11), KST62(11), KST63(11), KST64(11), KST65(11)                 2400
     A), KST66(11), KST67(11), KST68(11), KST69(11)                            2500
C                                                                              2600
      DATA KBL,KDIG/1H ,1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/               2700
      DATA KABC/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,       2800
     1 1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/                        2900
      DATA KSPK/1H=,1H,,1H(,1H/,1H),1H+,1H-,1H*,1H.,1H$,1H',1H:,1H"/           3000
      DATA MXCH,MXLI,N15B,LINE,NPAGE/6,8191,8192,51,0/                         3100
      DATA KAP/1HS,1HU,1HP,1HE,1HR,1H ,1HI,1HN,1HD,1HE,1HX/                    3200
      DATA KLAST,KSTOP/1HL,1HA,1HS,1HT,1HS,1HT,1HO,1HP/                        3300
C                                                                              3400
      DATA KST1/1H*,1H*,1HM,1HE,1HM,1HB,1HE,1HR,1H=,1H ,9/                     3500
      DATA KST2/1H/,1H/,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,2/                     3600
      DATA KST3/1HA,1HS,1HS,1HI,1HG,1HN,1H ,1H ,1H ,1H ,6/                     3700
      DATA KST4/1HB,1HA,1HC,1HK,1HS,1HP,1HA,1HC,1HE,1H ,9/                     3800
      DATA KST5/1HB,1HL,1HO,1HC,1HK,1HD,1HA,1HT,1HA,1H ,9/                     3900
      DATA KST6/1HB,1HU,1HF,1HF,1HE,1HR,1HI,1HN,1H(,1H ,9/                     4000
      DATA KST7/1HB,1HU,1HF,1HF,1HE,1HR,1HO,1HU,1HT,1H(,10/                    4100
      DATA KST8/1HC,1HA,1HL,1HL,1H ,1H ,1H ,1H ,1H ,1H ,4/                     4200
      DATA KST9/1HC,1HO,1HM,1HM,1HO,1HN,1H ,1H ,1H ,1H ,6/                     4300
      DATA KST10/1HC,1HO,1HM,1HP,1HL,1HE,1HX,1H ,1H ,1H ,7/                    4400
      DATA KST11/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE,1H ,1H ,8/                    4500
      DATA KST12/1HD,1HA,1HT,1HA,1H ,1H ,1H ,1H ,1H ,1H ,4/                    4600
      DATA KST13/1HD,1HE,1HF,1HI,1HN,1HE,1HF,1HI,1HL,1HE,10/                   4700
      DATA KST14/1HD,1HI,1HM,1HE,1HN,1HS,1HI,1HO,1HN,1H ,9/                    4800
      DATA KST15/1HD,1HO,1HU,1HB,1HL,1HE,1HP,1HR,1HE,1HC,10/                   4900
      DATA KST16/1HD,1HO,1HU,1HB,1HL,1HE,1H ,1H ,1H ,1H ,6/                    5000
      DATA KST17/1HD,1HO,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,2/                    5100
      DATA KST18/1HE,1HN,1HC,1HO,1HD,1HE,1H(,1H ,1H ,1H ,7/                    5200
      DATA KST19/1HE,1HN,1HD,1HF,1HI,1HL,1HE,1H ,1H ,1H ,7/                    5300
      DATA KST20/1HE,1HN,1HD,1H ,1H ,1H ,1H ,1H ,1H ,1H ,3/                    5400
      DATA KST21/1HE,1HN,1HT,1HR,1HY,1H ,1H ,1H ,1H ,1H ,5/                    5500
```

```
      DATA KST22/1HE,1HQ,1HU,1HI,1HV,1HA,1HL,1HE,1HN,1HC,10/                    5600
      DATA KST23/1HE,1HX,1HT,1HE,1HR,1HN,1HA,1HL,1H ,1H ,8/                     5700
      DATA KST24/1HF,1HI,1HN,1HD,1H ,1H ,1H ,1H ,1H ,1H ,4/                     5800
      DATA KST25/1HF,1HO,1HR,1HM,1HA,1HT,1H(,1H ,1H ,1H ,7/                     5900
      DATA KST26/1HF,1HO,1HR,1HT,1HR,1HA,1HN,1H ,1H ,1H ,7/                     6000
      DATA KST27/1HF,1HR,1HE,1HQ,1HU,1HE,1HN,1HC,1HY,1H ,9/                     6100
      DATA KST28/1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN,1H ,1H ,8/                     6200
      DATA KST29/1HG,1HO,1HT,1HO,1H(,1H ,1H ,1H ,1H ,1H ,5/                     6300
      DATA KST30/1HG,1HO,1HT,1HO,1H ,1H ,1H ,1H ,1H ,1H ,4/                     6400
      DATA KST31/1HI,1HF,1HA,1HC,1HC,1HU,1HM,1HU,1HL,1HA,10/                    6500
      DATA KST32/1HI,1HF,1HQ,1HU,1HO,1HT,1HI,1HE,1HN,1HT,10/                    6600
      DATA KST33/1HI,1HF,1H(,1HD,1HI,1HV,1HI,1HD,1HE,1HC,10/                    6700
      DATA KST34/1HI,1HF,1H(,1HE,1HN,1HD,1HF,1HI,1HL,1HE,10/                    6800
      DATA KST35/1HI,1HF,1H(,1HS,1HE,1HN,1HS,1HE,1HL,1HI,10/                    6900
      DATA KST36/1HI,1HF,1H(,1HS,1HE,1HN,1HS,1HE,1HS,1HW,10/                    7000
      DATA KST37/1HI,1HF,1H(,1H ,1H ,1H ,1H ,1H ,1H ,3/                         7100
      DATA KST38/1HI,1HN,1HT,1HE,1HQ,1HE,1HR,1H ,1H ,1H ,7/                     7200
      DATA KST39/1HL,1HO,1HG,1HI,1HC,1HA,1HL,1H ,1H ,1H ,7/                     7300
      DATA KST40/1HI,1HM,1HP,1HL,1HI,1HC,1HI,1HT,1H ,1H ,8/                     7400
      DATA KST41/1HN,1HA,1HM,1HE,1HL,1HI,1HS,1HT,1H ,1H ,8/                     7500
      DATA KST42/1HP,1HA,1HU,1HS,1HE,1H ,1H ,1H ,1H ,1H ,5/                     7600
      DATA KST43/1HP,1HR,1HI,1HN,1HT,1H ,1H ,1H ,1H ,1H ,5/                     7700
      DATA KST44/1HP,1HR,1HO,1HG,1HR,1HA,1HM,1H ,1H ,1H ,7/                     7800
      DATA KST45/1HP,1HU,1HN,1HC,1HH,1H ,1H ,1H ,1H ,1H ,5/                     7900
      DATA KST46/1HR,1HE,1HA,1HD,1HI,1HN,1HP,1HU,1HT,1HT,10/                    8000
      DATA KST47/1HR,1HE,1HA,1HD,1HT,1HA,1HP,1HE,1H ,1H ,8/                     8100
      DATA KST48/1HR,1HE,1HA,1HD,1H(,1H ,1H ,1H ,1H ,1H ,5/                     8200
      DATA KST49/1HR,1HE,1HA,1HD,1H ,1H ,1H ,1H ,1H ,1H ,4/                     8300
      DATA KST50/1HR,1HE,1HA,1HL,1H ,1H ,1H ,1H ,1H ,1H ,4/                     8400
      DATA KST51/1HR,1HE,1HT,1HU,1HR,1HN,1H ,1H ,1H ,1H ,6/                     8500
      DATA KST52/1HR,1HE,1HW,1HI,1HN,1HD,1H ,1H ,1H ,1H ,6/                     8600
      DATA KST53/1HS,1HE,1HG,1HM,1HE,1HN,1HT,1H ,1H ,1H ,7/                     8700
      DATA KST54/1HS,1HE,1HN,1HS,1HE,1HL,1HI,1HG,1HH,1HT,10/                    8800
      DATA KST55/1HS,1HT,1HO,1HP,1H ,1H ,1H ,1H ,1H ,1H ,4/                     8900
      DATA KST56/1HS,1HU,1HB,1HR,1HO,1HU,1HT,1HI,1HN,1HE,10/                    9000
      DATA KST57/1HT,1HY,1HP,1HE,1H ,1H ,1H ,1H ,1H ,1H ,4/                     9100
      DATA KST58/1HW,1HR,1HI,1HT,1HE,1HO,1HU,1HT,1HP,1HU,10/                    9200
      DATA KST59/1HW,1HR,1HI,1HT,1HE,1HT,1HA,1HP,1HE,1H ,9/                     9300
      DATA KST60/1HW,1HR,1HI,1HT,1HE,1H(,1H ,1H ,1H ,1H ,6/                     9400
      DATA KST61/1HC,1HH,1HA,1HR,1HA,1HC,1HT,1HE,1HR,1H , 9/                    9500
      DATA KST62/1HC,1HL,1HO,1HS,1HE,1H ,1H ,1H ,1H ,1H , 5/                    9600
      DATA KST63/1HE,1HL,1HS,1HE,1H ,1H ,1H ,1H ,1H ,1H , 6/                    9700
      DATA KST64/1HE,1HN,1HD,1HI,1HF,1H ,1H ,1H ,1H ,1H , 5/                    9800
      DATA KST65/1HI,1HN,1HQ,1HU,1HI,1HR,1HE,1H ,1H ,1H , 7/                    9900
      DATA KST66/1HI,1HN,1HT,1HR,1HI,1HN,1HS,1HI,1HC,1H , 9/                   10000
      DATA KST67/1HO,1HP,1HE,1HN,1H ,1H ,1H ,1H ,1H ,1H , 4/                   10100
      DATA KST68/1HP,1HA,1HR,1HA,1HM,1HE,1HT,1HE,1HR,1H , 9/                   10200
      DATA KST69/1HS,1HA,1HV,1HE,1H ,1H ,1H ,1H ,1H ,1H , 4/                   10300
      DATA IPAR, ICARD, L1, KLASS / 1, 1650*1H  /                              10400
      DATA OTAB / 63*Z26, Z25, 128*Z26,                                        10500
     C     Z01, Z02, Z03, Z04, Z05, Z06, Z07, Z08, Z09, 7*Z25,                 10600
     D     Z0A, Z0B, Z0C, Z0D, Z0E, Z0F, Z10, Z11, Z12, 7*Z25,                 10700
     E     Z25, Z13, Z14, Z15, Z16, Z17, Z18, Z19, Z1A, 6*Z25,                 10800
     F Z1B,Z1C, Z1D, Z1E, Z1F, Z20, Z21, Z22, Z23, Z24, 7*Z25  /               10900
        END                                                                    11000
```

```
      |      |SUBROUTINE CENTOR                                                                |  100
C     |      |                                                                                 |  200
C     |      |THIS ROUTINE CENTERS THE PAGE CAPTIONS.                                          |  300
C     |      |                                                                                 |  400
      |      |COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K               |  500
     1|      |RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM                     |  600
     2|      |  ,NERR,NERO,MEMKEY                                                              |  700
      |      |COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                                     |  800
      |      |COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE                                          |  900
      |      |COMMON /IOTA/ KEY(24)                                                            | 1000
      |      |COMMON /KAPPA/ KAP(11)                                                           | 1100
      |      |COMMON /OMEGA/ KLAST(4),KSTOP(4)                                                 | 1200
      |      |COMMON /SIGMA/ KSTIJ(11,60)                                                      | 1300
C     |      |                                                                                 | 1400
C     |      |COMPRESS STATEMENT BY ELIMINATING MULTIPLE BLANKS.                               | 1500
C     |      |                                                                                 | 1600
1     |      |DO 2 I=1,66                                                                      | 1700
      |      |  IF (JOB(I).NE.KBL) GO TO 3                                                     | 1800
2     |      |  CONTINUE                                                                       | 1900
      |      |RETURN                                                                           | 2000
C     |      |                                                                                 | 2100
3     |      |JOB(1)=JOB(I)                                                                    | 2200
   35 |      |J=1                                                                              | 2300
      |      |IB=I+1                                                                           | 2400
      |      |DO 4 I=IB,66                                                                     | 2500
      |      |  IF (JOB(I).EQ.KBL.AND.JOB(I-1).EQ.KBL) GO TO 4                                 | 2600
      |      |  J=J+1                                                                          | 2700
      |      |  JOB(J)=JOB(I)                                                                  | 2800
4     |      |  CONTINUE                                                                       | 2900
C     |      |                                                                                 | 3000
      |      |IB=J+1                                                                           | 3100
      |      |DO 5 I=IB,66                                                                     | 3200
5     |      |  JOB(I)=KBL                                                                     | 3300
C     |      |                                                                                 | 3400
C     |      |CENTER HEADING.                                                                  | 3500
C     |      |                                                                                 | 3600
      |      |IB=(66-J)/2                                                                      | 3700
6     |      |I=J+IB                                                                           | 3800
      |      |JOB(I)=JOB(J)                                                                    | 3900
      |      |J=J-1                                                                            | 4000
      |      |IF (J) 7,7,6                                                                     | 4100
C     |      |                                                                                 | 4200
C     |      |ELIMINATE REMAINING NON-BLANKS.                                                  | 4300
C     |      |                                                                                 | 4400
7     |      |IF (I.EQ.1) RETURN                                                               | 4500
      |      |IB=I-1                                                                           | 4600
      |      |DO 8 I=1,IB                                                                      | 4700
8     |      |  JOB(I)=KBL                                                                     | 4800
      |      |RETURN                                                                           | 4900
      |      |  END                                                                            | 5000
```

```
      SUBROUTINE ERREND                                                 100
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL  200
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                      300
      CALL ERROR                                                        400
      DO 15 I=KOL,KMAX                                                  500
        IF ( KARD(I) .EQ. KSPK(2) ) GO TO 25                            600
        IF ( KARD(I) .EQ. KSPK(5) ) GO TO 25                            700
   15   CONTINUE                                                        800
      RETURN                                                            900
   25 KOL = KOL-5                                                      1000
      N = I-KOL-1                                                      1100
      CALL MOVER( 0,DMMY,DMMY,N )                                      1200
      RETURN                                                           1300
        END                                                            1400
```

PAGE 1 OF ERREND

```
      SUBROUTINE ERROR                                                  100
      COMMON IS9685(9685),NERR,NERO,MEMKEY                              200
      COMMON /TAPES/ N1, N2, N3, N4, N5, N6                             300
C                                                                       400
      CALL PAG (1)                                                      500
1     WRITE (N6,2)                                                      600
      NERR = NERR+1                                                     700
      RETURN                                                            800
C                                                                       900
    2 FORMAT(' ## SYNTAX ##')                                           1000
      END                                                               1100
```

PAGE 1 OF ERROR

```
      SUBROUTINE ERR ( K,L,IFMT )                                                    100
      INTEGER*2 IFMT(2,1)                                                            200
      N6 = 6                                                                         300
      CALL PAG(1)                                                                    400
120   CONTINUE                                                                       500
C                1   2   3   4   5   6   7   8   9   10  11  12                     600
      GO TO ( 160,163,165,168,170,173,175,178,183,185,187,190                        700
     1   , 13, 14, 15, 16, 17, 18                                                    800
     2                                                          ), K                900
   13 WRITE (N6,9016)                                                              1000
      GO TO 140                                                                    1100
   14 WRITE (N6,9017) ( IFMT(I,1), I=1,4 )                                         1200
      GO TO 140                                                                    1300
   15 WRITE ( 6,9915)                                                             1400
      GO TO 140                                                                    1500
   16 WRITE (N6,89)                                                               1600
      GO TO 140                                                                    1700
   17 WRITE (6,205)                                                               1800
      GO TO 140                                                                    1900
   18 WRITE (6,9018) ( IFMT(I,1), I=1,L )                                         2000
      GO TO 140                                                                    2100
C                                                                                  2200
160   WRITE (N6,9003) IFMT(1,L)                                                    2300
      GO TO 140                                                                    2400
163   WRITE (N6,9004)                                                              2500
      GO TO 140                                                                    2600
165   WRITE (N6,9005)                                                              2700
      GO TO 140                                                                    2800
168   WRITE (N6,9006) IFMT(1,L)                                                    2900
      GO TO 140                                                                    3000
170   WRITE (N6,9007) IFMT(1,L)                                                    3100
      GO TO 140                                                                    3200
173   WRITE (N6,9008) IFMT(1,L)                                                    3300
      GO TO 140                                                                    3400
175   WRITE (N6,9009)                                                              3500
      GO TO 140                                                                    3600
178   WRITE (N6,9010)                                                              3700
      GO TO 140                                                                    3800
183   WRITE (N6,9012)                                                              3900
      GO TO 140                                                                    4000
185   WRITE (N6,9013)                                                              4100
      GO TO 140                                                                    4200
187   WRITE (N6,9014)                                                              4300
      GO TO 140                                                                    4400
190   WRITE (N6,9015) IFMT(1,L)                                                    4500
      GO TO 140                                                                    4600
C                                                                                  4700
140   CONTINUE                                                                     4800
      RETURN                                                                       4900
C                                                                                  5000
89    FORMAT (1HO,9X,5H******,5X,4(6H*******),5X,21HMISSING END STATEMENT,         5100
     1 5X,4(6H*******)/1X)                                                         5200
  205 FORMAT('0**  JOB TERMINATING FOR TOO MANY PARENS.')                          5300
9003  FORMAT(' CHARACTER (',A1,') IS NOT A VALID FORMAT TYPE.')                    5400
9004  FORMAT(' NO LEADING COUNT OF HOLLERITH CHARACTERS.')                         5500
```

```
 9005     FORMAT(' TWO SUCCESSIVE LETTERS.')                                5600
 9006     FORMAT(' P MULTIPLIER USED ON (',A1,') FORMAT.')                  5700
 9007     FORMAT(1XA1,' FORMAT NOT IN ANSI SUBSET LANGUAGE.')               5800
 9008     FORMAT(1XA1,' FORMAT NOT ANSI STANDARD.')                         5900
 9009     FORMAT(' NO LEADING COUNT ON (X) FORMAT.')                        6000
 9010     FORMAT(' Z FORMAT NOT ASA STANDARD.')                             6100
 9012     FORMAT(' NO LEADING COUNT ON L. PAREN IN FORMAT.')                6200
 9013     FORMAT(' MORE THAN ONE PERIOD IN A STRING.')                      6300
 9014     FORMAT(' TROUBLE IN AN APOST. HOLLERITH FORMAT.')                 6400
 9015     FORMAT(' A STRING ENDING WITH (',A1,') DID NOT HAVE ALL THE RIGHT 6500
     1     STUFF.')                                                          6600
 9016     FORMAT(' ARGUMENTS ON AN ENTRY STATEMENT ARE NOT ALLOWED IN THE CD6700
     ' C 6600 FORTRAN LANGUAGE.')                                          6800
 9017     FORMAT(1X4A2,'IS NOT A FORTRAN-SUPPLIED SUBPROGRAM ON THE CDC 6600
     ' ')                                                                  7000
 9018     FORMAT(' ## STATEMENT TYPE NOT ALLOWED IN IBM FORTRAN:  ',5A2)    7100
 9915     FORMAT( ' HEX')                                                   7200
          END                                                               7300
```

```
      SUBROUTINE FNCHEK( KEY )
      INTEGER COUNT1(40), COUNT2(38)
      LOGICAL*1 TEXL1(8,1), BLANK, STAR
C----- ---- ONLY FIRST 8 CHARACTERS ON NAME NEEDED FOR SUBPROGRAM TRANSL.
      REAL*8 NAME, TEXT1(55), TEXT2(38), SYMTAB
      COMMON ITYPE, SPACE(6675), SYMTAB(1)
      COMMON /HASH/ ILOC, LOC, NAME
      EQUIVALENCE ( TEXT1(1), TEXL1(1,1) )
      DATA BLANK, STAR / 1H , 1H* /
      DATA NFNS, NTR, NDT / 55, 40, 38 /
      DATA TEXT1   /
     1   8H DABS   , 8H DMAX1  , 8H DMIN1  , 8H DMOD   , 8H DFLOAT ,
     2   8H DEXP   , 8H DLOG   , 8H DLOG10 , 8H CDEXP  , 8H CDLOG  ,
     3   8H DARSIN , 8H ARSIN  , 8H DARCOS , 8H ARCOS  , 8H DATAN  ,
     4   8H DATAN2 , 8H DSIN   , 8H CDSIN  , 8H DCOS   , 8H CDCOS  ,
     5   8H DTAN   , 8H DSQRT  , 8H CDSQRT , 8H DTANH  , 8H CDABS  ,
     6   8H IDINT  , 8H HFIX   , 8H DSIGN  , 8H DCMPLX , 8H DCONJG ,
     7   8H LOG    , 8H LOG10  , 8H MAX    , 8H MIN    , 8H DINT   ,
     8   8H DDIM   , 8H DREAL  , 8H DIMAG  ,
     7   8H SNGL   , 8H DBLE   , 8H DCOTAN , 8H COTAN  , 8H DSINH  ,
     8   8H SINH   , 8H DCOSH  , 8H COSH   , 8H ERF    , 8H DERF   ,
     9   8H ERFC   , 8H DERFC  , 8H GAMMA  , 8H DGAMMA , 8H ALGAMA ,
     A   8H DLGAMA , 8H LGAMMA  /
      DATA TEXT2   /
     1   8H ABS    , 8H AMAX1  , 8H AMIN1  , 8H AMOD   , 8H FLOAT  ,
     2   8H EXP    , 8H ALOG   , 8H ALOG10 , 8H CEXP   , 8H CLOG   ,
     3   8H ASIN   , 8H ASIN   , 8H ACOS   , 8H ACOS   , 8H ATAN   ,
     4   8H ATAN2  , 8H SIN    , 8H CSIN   , 8H COS    , 8H CCOS   ,
     5   8H TAN    , 8H SQRT   , 8H CSQRT  , 8H TANH   , 8H CABS   ,
     6   8H INT    , 8H IFIX   , 8H SIGN   , 8H CMPLX  , 8H CONJG  ,
     7   8H ALOG   , 8H ALOG10 , 8H MAXO   , 8H MINO   , 8H AINT   ,
     8   8H DIM    , 8H REAL   , 8H AIMAG   /
      DATA COUNT1  /
     1              4,          5,          5,          4,          6,
     2              4,          4,          6,          5,          5,
     3              6,          5,          6,          5,          5,
     4              6,          4,          5,          4,          5,
     5              4,          5,          6,          5,          5,
     6              5,          4,          5,          6,          6,
     7              3,          5,          3,          3,          4,
     8              4,          5,          5,
     7              4,          4 /
      DATA COUNT2  /
     1              3,          5,          5,          4,          5,
     2              3,          4,          6,          4,          4,
     3              4,          4,          4,          4,          4,
     4              5,          3,          4,          3,          4,
     5              3,          4,          5,          4,          4,
     6              3,          4,          4,          5,          5,
     7              4,          6,          4,          4,          4,
     8              3,          4,          5 /
      IF ( KEY .EQ. 0 ) GO TO 101
      DO 11 I=1,NFNS
        IF ( NAME .EQ. TEXT1(I) ) GO TO 21
   11   CONTINUE
```

```
      RETURN
C---- MATCH FOUND.  IF UNTRANSLATABLE, PRINT OUT ERROR MESSAGE
   21 IF ( KEY .GT. 0 ) GO TO 25
C---- INTRINSIC FUNCTION NAME FOUND IN EXTERNAL STATEMENT.  FLAG
C         IN TEXT1 SO THAT NO CONVERSION TAKES PLACE.
      TEXL1(1,I) = STAR
      RETURN
   25 IF ( I .GT. NTR ) GO TO 41
      IF ( I .GT. NDT ) GO TO 31
      NAME = TEXT2(I)
      CALL MOVER( COUNT2(I)+1,NAME,1,COUNT1(I) )
      IF ( SYMTAB(ILOC) .NE. TEXT1(I) ) RETURN
C---- REMOVE OLD NAME FROM SYMBOL TABLE
      ILOC = ILOC-1
      CALL UPDATE
      RETURN
C---- ELIMINATE FUNCTION NAME DBLE OR SNGL.
   31 K = KOL
      KOL = K-5
      CALL MOVER( 0,DMMY,DMMY,COUNT1(I) )
      KOL = K
      IF ( SYMTAB(ILOC) .EQ. TEXT1(I) ) ILOC = ILOC-1
      RETURN
C---- UNTRANSLATABLE NAME
   41 CALL ERR( 14,DMMY,NAME )
      RETURN
C---- INITIALIZE
  101 DO 111 I=1,NFNS
  111   TEXL1(1,I) = BLANK
      RETURN
        END
```

```
      SUBROUTINE FORTRN ( KMAX, IFMT, DTEFLG )                          100
C                                                                       200
C-----  ---- 360 TO 6600 FORMAT TRANSLATOR.                             300
C                                                                       400
      INTEGER*2 IFMT(2,1), LEE, LHH, KBL, INO, IANS, LPAREN, RPAREN     500
      INTEGER*2 KEYCHR, JCARD                                           600
      LOGICAL*4 DTEFLG                                                  700
      LOGICAL*1         LLEAD, LPMLT, LLETR, LPERD, LCOMA, EFGFLG       800
      LOGICAL*1 JHCOM                                                   900
      DIMENSION IANS(4), INO(10)                                       1000
      COMMON / DELTA/ MOVE, JCARD(2,80), ICARD(1600), IOL, NCARD       1100
     +              , OSTMTN, FRSTSB, IC, JC                           1200
      DATA LEE / 1HE /, LHH / 1HH /, KBL / 1H  /                       1300
      DATA INO / 1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9 /             1400
C                                                                      1500
C-----  ---- INITIALIZATION.                                           1600
C                                                                      1700
      CALL ICHAR( IFMT, KMAX )                                         1800
      K = 0                                                            1900
      L = 0                                                            2000
      ASSIGN 120 TO MGO                                                2100
      ASSIGN 20 TO NGO                                                 2200
      LPAREN = 0                                                       2300
      RPAREN = 0                                                       2400
      EFGFLG = .FALSE.                                                 2500
      N6 = 6                                                           2600
C                                                                      2700
C-----  ---- MOVE FORMAT STATEMENT NO.  (ASSUME NO. EXISTS.)           2800
C                                                                      2900
      DO 10 I=1,6                                                      3000
        K = K + 1                                                      3100
        L = L + 1                                                      3200
   10   JCARD(1,K) = IFMT(1,L)                                         3300
C                                                                      3400
C-----  ---- ASSUME THE STRING "FORMAT(" EXISTS.                       3500
C                                                                      3600
   20   L = L + 1                                                      3700
   30   IF ( IFMT(2,L) .EQ. 48 ) GO TO 20                              3800
        IF ( IFMT(2,L) .EQ. 39 ) ASSIGN 50 TO NGO                      3900
        ASSIGN 40 TO MGO                                               4000
        GO TO 700                                                      4100
40      GO TO NGO, ( 20,50 )                                           4200
   50   LPAREN = LPAREN + 1                                            4300
        GO TO 126                                                      4400
C                                                                      4500
C-----  ---- A STRING OF SOMETHING HAS BEEN COMPLETED.  DID IT CONTAIN 4600
C-----  ---- ALL THE RIGHT THINGS.                                     4700
C                                                                      4800
  120   IF ( JHCOM .AND. LCOMA ) GO TO 126                             4900
        IF ( .NOT. LLEAD ) GO TO 190                                   5000
        IF ( .NOT. LPMLT ) GO TO 190                                   5100
        IF ( .NOT. LLETR ) GO TO 190                                   5200
        IF ( .NOT. LPERD ) GO TO 190                                   5300
        IF ( .NOT. LCOMA ) GO TO 190                                   5400
  126   CONTINUE                                                       5500
```

```
      ASSIGN 140 TO MGO
      KOUNT = 0
C
C---- ---- PREPARE FOR A STRING OF SOMETHING.  WE WILL WANT TO CHECK
C---- ---- AT THE END OF THE STRING TO SEE THAT NO ILLEGAL STRING WAS
C---- ---- ENCOUNTERED.
C
  130 LLEAD = .FALSE.
      LPMLT = .FALSE.
      LLETR = .FALSE.
      LPERD = .FALSE.
      LCOMA = .FALSE.
      JHCOM = .TRUE.
      GO TO 150
C
C---- ---- PICK UP NEXT CHARACTER.
C
  140 CONTINUE
      JHCOM = .FALSE.
  150 L = L + 1
      IF ( L .GT. KMAX ) GO TO 730
  151 KOG = IFMT(2,L)
C                A   B   C   D   E   F   G   H   I   J   K   L   M
      GO TO ( 210,170,160,240,238,240,230,280,210,160,160,210,160,
C                N   O   P   Q   R   S   T   U   V   W   X   Y   Z
     1          160,178,360,160,160,250,260,160,160,160,440,160,178,
C                0   1   2   3   4   5   6   7   8   9   =   ,   (
     2          470,470,470,470,470,470,470,470,470,470,160,580,590,
C                /   )   +   -   *   .   $   '  BL NONE
     3          600,610,160,160,670,650,160,670,150,160   ) , KOG
C
C---- ---- B,C,J,K,M,N,O,Q,R,S,T,U,V,W,Y,Z,=,+,-,*,$,NO MATCH, OR ERROR.
C
160   KERR = 1
      GO TO 195
164   KERR = 2
      GO TO 195
165   KERR = 3
      GO TO 195
168   KERR = 4
      GO TO 195
  170 IF ( IFMT(2,L+1) .EQ. 14  .OR.  IFMT(2,L+1) .EQ. 26 ) GO TO 220
173   KERR = 6
      GO TO 195
175   KERR = 7
      ASSIGN 126 TO MGO
      GO TO 195
178   KERR = 8
      GO TO 195
183   KERR = 9
      GO TO 195
185   KERR = 10
      GO TO 195
187   KERR = 11
      GO TO 195
```

PAGE 2 OF FORTRN

```
190     KERR = 12                                                          11100
        CALL ERR ( KERR,L,IFMT )                                           11200
        GO TO 126                                                          11300
C                                                                          11400
195     CALL ERR ( KERR,L,IFMT )                                           11500
        LLEAD = .TRUE.                                                     11600
        LPMLT = .TRUE.                                                     11700
        LLETR = .TRUE.                                                     11800
        LPERD = .TRUE.                                                     11900
        LCOMA = .TRUE.                                                     12000
        GO TO 700                                                          12100
C                                                                          12200
C-------- A FORMAT.  LEADING NO. OK.  LAGGING NO. OK.  LEAD LETTER NOT.    12300
C                                                                          12400
  210   IF ( LLETR ) GO TO 165                                             12500
        IF ( LPMLT ) GO TO 168                                             12600
        LLEAD = .TRUE.                                                     12700
        LPMLT = .TRUE.                                                     12800
        LLETR = .TRUE.                                                     12900
        LPERD = .TRUE.                                                     13000
        GO TO 700                                                          13100
C-------- TEST FOR BN OR BZ                                                13200
  220   ASSIGN 225 TO MGO                                                  13300
        GO TO 700                                                          13400
  225   ASSIGN 126 TO MGO                                                  13500
        GO TO 700                                                          13600
C                                                                          13700
C-------- G FORMAT; NOT IN ANSI SUBSET                                     13800
  230   IF ( LLETR ) GO TO 165                                             13900
        KERR = 5                                                           14000
        CALL ERR( KERR,L,IFMT )                                            14100
        IFMT(1,L) = LEE                                                    14200
C-------- E.  TEST IF OF FORM EW.DEE                                       14300
  238   IF ( .NOT. LLETR ) GO TO 242                                       14400
        IF ( KE.EQ.4 .OR. KE.EQ.5 .OR. KE.EQ.7 ) GO TO 239                 14500
        GO TO 178                                                          14600
  239   IF ( .NOT. LPERD  .OR.  NE .GT. 1 ) GO TO 178                      14700
        NE = NE + 1                                                        14800
        GO TO 700                                                          14900
C-------- D FORMAT.  LEADING  P  OK.  LEADING NO. OK.  LAGGING NO. OK.     15000
C--------           LATER PERIOD REQUIRED.  LEADING LETTER NOT OK.         15100
C                                                                          15200
  240   IF ( LLETR ) GO TO 165                                             15300
  242   NE = 0                                                             15400
        KE = KOG                                                           15500
        LPMLT = .TRUE.                                                     15600
        LLEAD = .TRUE.                                                     15700
        LLETR = .TRUE.                                                     15800
        LPERD = .FALSE.                                                    15900
        LCOMA = .FALSE.                                                    16000
        GO TO 700                                                          16100
  250   CONTINUE                                                           16200
  260   CONTINUE                                                           16300
        GO TO 160                                                          16400
C                                                                          16500
```

```
C----  -  ---- H FORMAT.  MOVE 'KOUNT' CHARACTERS.                          16600
C                                                                          16700
  280     IF ( LLETR ) GO TO 165                                           16800
          IF ( .NOT. LLEAD ) GO TO 164                                     16900
          IF ( LPMLT ) GO TO 168                                           17000
          LCOMA = .TRUE.                                                   17100
          ASSIGN 281 TO MGO                                                17200
          GO TO 700                                                        17300
2810      K = K - 1                                                        17400
281       CONTINUE                                                         17500
          K = K + 1                                                        17600
          L = L + 1                                                        17700
282       IF ( IFMT(1,L) .EQ. 47 ) GO TO 2810                              17800
  283     JCARD(1,K) = IFMT(1,L)                                           17900
          KOUNT = KOUNT - 1                                                18000
          IF ( KOUNT .GT. 0 ) GO TO 281                                    18100
          GO TO 126                                                        18200
C                                                                          18300
C----  -  ---- P FORMAT.  LEADING NO. OK.  A D,E,F, OR G MUST EVENTUALLY   18400
C----  -  ----              FOLLOW.                                       18500
C                                                                          18600
  360     LPMLT = .TRUE.                                                   18700
          LCOMA = .FALSE.                                                  18800
          GO TO 700                                                        18900
C                                                                          19000
C----  -  ---- X FORMAT.  LEADING NUMBER MUST EXIST. LEADING LETTER NOT OK. 19100
C                                                                          19200
  440     IF ( LLETR ) GO TO 165                                           19300
          IF ( .NOT. LLEAD ) GO TO 175                                     19400
          IF ( LPMLT ) GO TO 168                                           19500
          LLETR = .TRUE.                                                   19600
          LCOMA = .TRUE.                                                   19700
          ASSIGN 126 TO MGO                                                19800
          GO TO 700                                                        19900
C                                                                          20000
C----  -  ---- A NUMBER.  CAN I HAVE A NUMBER NOW.                         20100
C                                                                          20200
  470     IF ( LLEAD ) GO TO 474                                           20300
  472     LLEAD = .TRUE.                                                   20400
          KOUNT = IFMT(2,L) - 27                                           20500
          GO TO 700                                                        20600
  474     IF ( LLETR ) GO TO 700                                           20700
          IF ( LPMLT ) GO TO 472                                           20800
          KOUNT = 10*KOUNT + IFMT(2,L) - 27                                20900
          GO TO 700                                                        21000
C                                                                          21100
C----  -  ---- A COMMA.  CAN WE HAVE A COMMA HERE.                         21200
C                                                                          21300
  580     CONTINUE                                                         21400
  583     LCOMA = .TRUE.                                                   21500
          ASSIGN 120 TO MGO                                                21600
          GO TO 700                                                        21700
C                                                                          21800
C----  -  ---- L. PAREN.  MUST BE PRECEEDED BY A NUMBER.                   21900
C                                                                          22000
```

```
  590  CONTINUE                                                          22100
       LPAREN = LPAREN + 1                                               22200
       LLEAD = .TRUE.                                                    22300
       LPMLT = .TRUE.                                                    22400
       LLETR = .TRUE.                                                    22500
       LPERD = .TRUE.                                                    22600
       GO TO 583                                                         22700
C                                                                        22800
C-----  ---- A SLASH.  SLASHES CAN COME ANYWHERE COMMAS CAN.             22900
C                                                                        23000
  600  CONTINUE                                                          23100
       GO TO 583                                                         23200
C                                                                        23300
C-----  ---- R. PAREN.  LIKE A SLASH OR A COMMA.                        23400
C                                                                        23500
  610  RPAREN = RPAREN + 1                                               23600
       GO TO 583                                                         23700
C                                                                        23800
C-----  ---- A PERIOD.  CAN I HAVE A PERIOD HERE.                       23900
C                                                                        24000
  650  IF ( LPERD ) GO TO 185                                            24100
       LPERD = .TRUE.                                                    24200
       GO TO 700                                                         24300
C                                                                        24400
C-----  ---- AN APOST.  SHIFT INTO HOLLERITH.                           24500
C                                                                        24600
  670  IF ( LLEAD ) GO TO 187                                            24700
       IF ( LPMLT ) GO TO 187                                            24800
       IF ( LLETR ) GO TO 187                                            24900
       IF ( LPERD ) GO TO 187                                            25000
       KEYCHR = IFMT(1,L)                                                25100
  671  CONTINUE                                                          25200
       KOUNT = 0                                                         25300
       KK = L                                                            25400
  673  KOUNT = KOUNT + 1                                                 25500
       IF ( KOUNT .GT. 1680 ) GO TO 677                                  25600
       KK = KK + 1                                                       25700
  674  IF ( IFMT(1,KK) .NE. KEYCHR ) GO TO 673                           25800
C-----  ---- SECOND APOST. ENCOUNTERED.                                  25900
       KSAV = KK                                                         26000
       KK = KK + 1                                                       26100
  675  IF ( IFMT(1,KK) .NE. KEYCHR ) GO TO 676                           26200
C-----  ---- DOUBLE APOST.                                               26300
       IFMT(1,KK) = 47                                                   26400
       IFMT(2,KK) = 48                                                   26500
       GO TO 673                                                         26600
  676  KK = KSAV                                                         26700
       IFMT(1,KK) = KBL                                                  26800
       IFMT(2,KK) = 48                                                   26900
       GO TO 678                                                         27000
  677  CALL PAG ( 1 )                                                    27100
       WRITE (N6,9037)                                                   27200
  678  KOUNT = KOUNT - 1                                                 27300
       NO = KOUNT                                                        27400
C                                                                        27500
```

```
C-----|-|---- BUILD '35H' IF KOUNT IS 35.                                    27600
C     | |                                                                   27700
      | |DO 680 I=1,4                                                       27800
      | |  MPY = 10**(4-I)                                                  27900
      | |  NMOD = NO / MPY                                                  28000
      | |  IANS(I) = INO(NMOD+1)                                            28100
680   | |  NO = NO - NMOD*MPY                                               28200
      | |DO 682 I=1,4                                                       28300
      | |  IF ( IANS(I) .NE. INO(1) ) GO TO 683                             28400
682   | |  IANS(I) = KBL                                                    28500
683   | |DO 686 I=1,4                                                       28600
      | |  IF ( IANS(I) .EQ. KBL ) GO TO 686                                28700
      | |  K = K + 1                                                        28800
 684  | |  JCARD(1,K) = IANS(I)                                             28900
686   | |  CONTINUE                                                         29000
      | |K = K + 1                                                          29100
      | |JCARD(1,K) = LHH                                                   29200
      | |GO TO 281                                                          29300
C     | |                                                                   29400
C-----|-|---- BUMP K.  STORE A CHARACTER.  GO GET THE NEXT ONE.              29500
C     | |                                                                   29600
 698  | |EFGFLG = .FALSE.                                                   29700
 700  | |K = K + 1                                                          29800
      | |JCARD(1,K) = IFMT(1,L)                                             29900
      | |GO TO MGO, ( 120,126,140,40,281 )                                  30000
C     | |                                                                   30100
C-----|-|---- ALL CARDS HAVE BEEN SCANNED.  WRAP IT UP AND GO HOME.         30200
C     | |                                                                   30300
 730  | |IF ( LPAREN .EQ. RPAREN ) GO TO 740                                30400
      | |CALL PAG ( 3 )                                                     30500
      | |WRITE (N6,9030) LPAREN, RPAREN                                     30600
 740  | |CONTINUE                                                           30700
      | |JC = K + 1                                                         30800
      | |MOVE = 1                                                           30900
      | |RETURN                                                             31000
9030  | |FORMAT(19HONO. OF L. PARENS =I2 / 19HONO. OF R. PARENS =I2 )       31100
9037  | |FORMAT(' SOMETHING WRONG WITH APOST. HOLLERITH FMT.')              31200
      | |  END                                                              31300
```

```
         SUBROUTINE IMPSET( IDX )
         INTEGER COMMA, RPAREN
         LOGICAL OUPDAT
         COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K
       1 RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM
       2  ,NERR,NERO,MEMKEY
         COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)
         COMMON / HASH / ILOC, LOC, NAME, IVAL
         COMMON /XUPDAT/ OUPDAT, ICDICT(27)
         DATA COMMA, RPAREN  / 1H,, 1H)  /
         OUPDAT = .TRUE.
         IFLAG = IABS( ITYPE )
         IF ( IFLAG .GT. 18 ) GO TO 25
         GO TO ( 25,25, 3,25, 5, 6,25, 8,25,25,25,12,13,25,25,25,25,18),
       *         IFLAG
    3    KEY = IVAL + 5
         GO TO 50
    6    KEY = 2
         GO TO 50
   12    KEY = 3
         GO TO 50
   13    KEY = 4
         GO TO 50
   18    KEY = 5
         GO TO 50
    5    CONTINUE
    8    CONTINUE
   25    KEY = 0
   50    CONTINUE
C----- ---- FIND FIRST CHARACTER
   95    DO 96 I=1,26
           IF ( KARD(KOL) .EQ. KABC(I) ) GO TO 97
   96      CONTINUE
         GO TO 105
C----- ---- FIND SECOND CHARACTER
   97    IF ( KARD(KOL+1) .EQ. KSPK(7) ) GO TO 970
         J = I
         GO TO 99
  970    KOL = KOL+2
         DO 98 J=1,26
           IF ( KARD(KOL) .EQ. KABC(J) ) GO TO 99
   98      CONTINUE
         GO TO 105
   99    DO 100 K=I,J
  100      ICDICT(K) = KEY
C----- ---- TEST FOR END OF STATEMENT
         IF ( KARD(KOL+1) .NE. COMMA ) GO TO 101
         KOL = KOL+2
         GO TO 95
  101    IF ( KARD(KOL+1) .NE. RPAREN) GO TO 105
         IF ( KOL+2 .GE. KMAX ) GO TO 120
         KOL = KOL+2
         IDX = 1
         GO TO 150
C----- ---- ERRORS
```

PAGE 1 OF IMPSET

```
105   CALL ERROR                                                        5600
      OUPDAT = .FALSE.                                                  5700
120   IDX = 2                                                           5800
150   CONTINUE                                                          5900
      IF ( OUPDAT ) CALL SETTMP                                         6000
      RETURN                                                            6100
        END                                                             6200
```

```
      SUBROUTINE INPUT                                                          100
C                                                                               200
C     THIS SUBROUTINE READS ALL INPUT CARDS AND CHECKS THEM FOR COMMENTS        300
C     AND CONTROL STATEMENTS.                                                   400
C                  THE STATEMENT IS PACKED IN THE SUPER-CARD ARRAY,             500
C     KARD.  KMAX IS THEN SET TO THE NUMBER OF COLUMNS IN KARD THAT             600
C     CONTAIN VALID DATA.                                                       700
C                                                                               800
      LOGICAL*1 OSEQ2(8), OCHAR                                                 900
      LOGICAL OSTMTN, FRSTSB, KSBO, KCDO, YES                                  1000
      REAL*8 SEQ1, SEQ2, SEQ3, SEQ0                                            1100
      COMMON /SEQ/ SEQ1, SEQ2, SEQ3                                            1200
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K       1300
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM             1400
     2  ,NERR,NERO,MEMKEY                                                     1500
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                             1600
      COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE,KSBO, KCDO,KOUT(80)              1700
      COMMON / DELTA/ MOVE, JCARD(80), ICARD(1600), IOL, NCARD                  1800
     +              , OSTMTN, FRSTSB, IC, JC                                   1900
      COMMON /KAPPA/ KAP(11)                                                   2000
      COMMON /OMEGA/ KLAST(4),KSTOP(4)                                         2100
      COMMON /SIGMA/ KSTIJ(11,60)                                              2200
      COMMON /TAPES/ N1, N2, N3, N4, N5, N6                                    2300
      EQUIVALENCE (KBUFF(1),KB1), (KBUFF(6),KB6)                               2400
      EQUIVALENCE ( OSEQ2(1),SEQ2 ), ( OCHAR,ICHAR )                           2500
      DATA SEQ0 / 8H00000000 /                                                 2600
C                                                                              2700
CC    A    B    C    D    E    F    G    H    I    J    K    L    M           2800
C     1    2    3    4    5    6    7    8    9    10   11   12   13          2900
C                                                                              3000
C     N    O    P    Q    R    S    T    U    V    W    X    Y    Z           3100
C     14   15   16   17   18   19   20   21   22   23   24   25   26          3200
C                                                                              3300
C     =    ,    (    /    )    +    -    *    .    $    '                     3400
C     1    2    3    4    5    6    7    8    9    10   11                    3500
C                                                                              3600
C     CHECK FOR FIRST CALL.                                                    3700
      KCDO = .TRUE.                                                            3800
1     IF (NPAGE) 2,2,4                                                         3900
C                                                                              4000
C     THIS IS FIRST CALL.  LOAD EMPTY BUFFER, KBUFF.                           4100
2     CONTINUE                                                                 4200
      CALL READ( KBUFF, &51)                                                   4300
      GO TO 52                                                                 4400
   51 CALL STARL ( KBUFF )                                                     4500
   52 NPAGE = 1                                                                4600
      SEQ1 = SEQ0                                                              4700
      SEQ3 = SEQ0                                                              4800
      DO 3 I=1,72                                                              4900
        IF (KBUFF(I).NE.KBL) GO TO 44                                          5000
3       CONTINUE                                                               5100
      GO TO 2                                                                  5200
    4 SEQ1 = SEQ2                                                              5300
      IF ( MOVE .LT. 0 ) GO TO 44                                              5400
      IF ( MOVE .EQ. 0 ) GO TO 411                                             5500
```

```
      KOL = MIN0( KOL, KMAX )                                                   5600
      IF ( MOVE .EQ. 1 ) CALL MOVER( 0,DMMY,DMMY,0 )                            5700
      JC = JC - 1                                                               5800
      CALL TRANCE( JCARD,JC )                                                   5900
      GO TO 44                                                                  6000
  411 KOL = KOL - 1                                                             6100
      IF ( KRSX .NE. KSPK(10) ) KOL = KMAX                                      6200
      CALL TRANCE( KARD,KOL )                                                   6300
   44 MOVE = 0                                                                  6400
      JC = 0                                                                    6500
      NROUT = 0                                                                 6600
      FRSTSB = .TRUE.                                                           6700
      IF ( KRSX .EQ. KSPK(10) ) GO TO 42                                        6800
      IF ( NREC ) 5,5,8                                                         6900
C                                                                               7000
C     SET UP PAGE CAPTION.                                                      7100
C                                                                               7200
5     IF (KBUFF(1).EQ.KSPK(8)) GO TO 360                                        7300
   55 CALL EQUAT4( 66,JOB,KBUFF(7) )                                            7400
      CALL CENTOR                                                               7500
      GO TO 8                                                                   7600
C                                                                               7700
C     COPY THROUGH COMMENTS, LOOKING FOR FORTRAN STATEMENTS.                    7800
C                                                                               7900
    7 CONTINUE                                                                  8000
      CALL TRANCE( KBUFF,72 )                                                   8100
      CALL READ( KBUFF, &61)                                                    8200
      GO TO 62                                                                  8300
   61 CALL STARL ( KBUFF )                                                      8400
   62 CONTINUE                                                                  8500
8     IF (KB1.EQ.KABC(3).OR.KB1.EQ.KSPK(10)) GO TO 7                            8600
      IF (KB1.EQ.KSPK(8)) GO TO 37                                              8700
C---- FLAG CONTINUATION CARDS INTERSPERSED WITH COMMENTS.                       8800
      IF ( KB6 .EQ. KBL  .OR.  KB6 .EQ. KDIG(1) ) GO TO 80                      8900
      CALL ERROR                                                                9000
C                                                                               9100
C     HAVE FORTRAN STATEMENT.  START PACKING KARD.                              9200
C                                                                               9300
   80 NREC=NREC+1                                                               9400
      NCARD = NCARD+1                                                           9500
      CALL EQUAT4( 80,KOUT,KBUFF )                                              9600
      CALL EQUAT4( 72,KARD,KBUFF )                                              9700
      K7=7                                                                      9800
      K72=72                                                                    9900
C                                                                              10000
C     LOOK FOR CONTINUATION CARDS AND PACK IN KARD.                            10100
C                                                                              10200
      DO 13 J=2,20                                                             10300
   10   CALL READ( KBUFF, &71)                                                 10400
        GO TO 72                                                               10500
   71   CALL STARL ( KBUFF )                                                   10600
   72   CONTINUE                                                               10700
        IF ( KBUFF(1) .NE. KABC(3) ) GO TO 105                                 10800
C---- ELIMINATE COMMENT CARDS INTERSPERSED WITH CONTINUATIONS.                 10900
        CALL TRANCE( KBUFF,72 )                                                11000
```

```
      GO TO 10                                                                  11100
  105 DO 11 I=1,72                                                              11200
        IF (KBUFF(I).NE.KBL) GO TO 12                                           11300
11      CONTINUE                                                                11400
      GO TO 10                                                                  11500
C                                                                               11600
   12 IF (KB1.EQ.KABC(3).OR.KB1.EQ.KSPK(4).OR.KB1.EQ.KSPK( 8).OR.KB6.EQ.        11700
     1KBL.OR.KB6.EQ.KDIG(1)) GO TO 16                                           11800
      K7=K7+66                                                                  11900
      CALL EQUAT4( 66,KARD(K7),KBUFF(7) )                                       12000
      L = 72                                                                    12100
      DO 13 I=1,8                                                               12200
        L = L+1                                                                 12300
        ICHAR = KBUFF(L)                                                        12400
   13   OSEQ2(I) = OCHAR                                                        12500
C                                                                               12600
C     NINETEEN CONTINUATION CARDS.  LOAD EMPTY BUFFER.                          12700
C                                                                               12800
   14 CALL READ( KBUFF, &81)                                                    12900
      GO TO 82                                                                  13000
   81 CALL STARL ( KBUFF )                                                      13100
   82 CONTINUE                                                                  13200
      DO 15 I=1,72                                                              13300
        IF (KBUFF(I).NE.KBL) GO TO 16                                           13400
15      CONTINUE                                                                13500
      GO TO 14                                                                  13600
C                                                                               13700
C     GET STATEMENT NUMBER, IF ANY.                                             13800
C                                                                               13900
16    KODE=0                                                                    14000
      OSTMTN = .FALSE.                                                          14100
      DO 19 I=1,5                                                               14200
        IF (KARD(I).EQ.KBL) GO TO 19                                            14300
        DO 17 J=1,10                                                            14400
          IF (KARD(I).EQ.KDIG(J)) GO TO 18                                      14500
17        CONTINUE                                                              14600
        IF ( KARD(1) .NE. KSPK(4) ) GO TO 20                                    14700
C-----  ---- JCL.  SKIP CARD                                                    14800
        CALL PAG( 1 )                                                           14900
        WRITE (6,170)                                                           15000
  170   FORMAT(10X25H**  JCL CARD DELETED.  **)                                 15100
        GO TO 44                                                                15200
18      IF (J.EQ.1.AND.KODE.EQ.0) GO TO 19                                      15300
        OSTMTN = .TRUE.                                                         15400
        GO TO 20                                                                15500
19      CONTINUE                                                                15600
C                                                                               15700
C     SQUEEZE OUT ALL BLANKS.  PUT BLANK IN KARD(1), AND START SQUEEZE          15800
C     IN KARD(2).                                                               15900
C                                                                               16000
   20 KMAX = K7 + 65                                                            16100
      KARD(6)=KBL                                                               16200
36    CONTINUE                                                                  16300
      RETURN                                                                    16400
C                                                                               16500
```

```
C       |  |LOOK FOR #LAST OR #STOP CONTROL CARDS.                                  |  16600
C       |  |                                                                       |  16700
  360   |  |NEWSUB = 1                                                             |  16800
        |  |GO TO 370                                                              |  16900
   37   |  |NEWSUB = 0                                                             |  17000
  370   |  |I=1                                                                    |  17100
        |  |DO 38 J=2,72                                                           |  17200
        |  |  IF (KBUFF(J).EQ.KBL) GO TO 38                                        |  17300
        |  |  IF (KBUFF(J).NE.KLAST(I)) GO TO 39                                   |  17400
        |  |  I=I+1                                                                |  17500
        |  |  IF (I.EQ.5) GO TO 41                                                 |  17600
38      |  |  CONTINUE                                                             |  17700
        |  |GO TO 400                                                              |  17800
39      |  |I=1                                                                    |  17900
        |  |DO 40 J=2,72                                                           |  18000
        |  |  IF (KBUFF(J).EQ.KBL) GO TO 40                                        |  18100
        |  |  IF (KBUFF(J).NE.KSTOP(I)) GO TO 400                                  |  18200
        |  |  I=I+1                                                                |  18300
        |  |  IF (I.EQ.5) GO TO 41                                                 |  18400
40      |  |  CONTINUE                                                             |  18500
  400   |  |IF ( NEWSUB ) 7,7,55                                                   |  18600
C       |  |                                                                       |  18700
C       |  |                    ######################                             |  18800
C       |  |                    ######################                             |  18900
41      |  |CONTINUE                                                               |  19000
        |  |CALL CLOSE                                                             |  19100
        |  |WRITE (N6,46)                                                          |  19200
        |  |STOP                                                                   |  19300
C       |  |                                                                       |  19400
C-----  -  |---- MULTIPLE STATEMENTS ON ONE EXTENDED CARD IMAGE                     |  19500
   42   |  |KOL = KOL + 2                                                          |  19600
        |  |KMAX = KMAX - KOL + 1                                                  |  19700
        |  |KARD(1) = KBL                                                          |  19800
        |  |CALL EQUAT4( 5, KARD(2), KARD(1) )                                     |  19900
        |  |CALL EQUAT4( KMAX, KARD(7), KARD(KOL) )                                |  20000
        |  |GO TO 36                                                               |  20100
   46   |  |FORMAT('0 END OF TRANSLATION')                                         |  20200
        |  |  END                                                                  |  20300
```

```
      |     |SUBROUTINE INSRTR( L1,TEXT,NCHAR,KEY )                                  |  100
      |     |INTEGER BLANK, EQUAL, JCARD(80), PCARD, PMAX                            |  200
      |     |LOGICAL OSTMTN, FRSTSB                                                  |  300
      |     |LOGICAL*1 TEXT(1), OCHAR, OSEQ1(8)                                      |  400
      |     |REAL*8 SEQ1, SEQ2, SEQ3                                                 |  500
      |     |COMMON /SEQ/ SEQ1, SEQ2, SEQ3                                           |  600
      |     |COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K      |  700
      |   1 |RSX,LIST(8191),LOCN,MINA,MAXA,NCHXX,NEXT,NRT,NREC,NROUT,NSYM            |  800
      |   2 |  ,NERR,NERO,MEMKEY                                                     |  900
      |     |COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                            | 1000
      |     |COMMON / DELTA/ MOVE, PCARD(80), ICARD(1600), IOL, NCARD                | 1100
      |   1 |   , OSTMTN, FRSTSB, IC, JC                                             | 1200
      |     |COMMON /TAPES/ N1, N2, N3, N4, N5, N6                                   | 1300
      |     |EQUIVALENCE ( OCHAR,ICHAR ), ( BLANK,KBL ), ( OSEQ1(1), SEQ1 )          | 1400
C-----|-    |---- GENERATE SEPARATE FORTRAN STATEMENT FOR DELETED TEXT                | 1500
C-----|-    |---- IF THE ORIGINAL STATEMENT HAD A STATEMENT NUMBER AND IF             | 1600
C     |     |     THIS IS THE FIRST NEW STATEMENT GENERATED FROM THE ORIGINAL        | 1700
C     |     |     STATEMENT, PUT THE STATEMENT NUMBER FIELD INTO THIS CARD.          | 1800
C     |     |     OTHERWISE, BLANK OUT STATEMENT NUMBER FIELD.                       | 1900
      |     |IF ( OSTMTN .AND. FRSTSB ) GO TO 20                                     | 2000
      |     |DO 10 I=1,6                                                             | 2100
   10 |     |  JCARD(I) = BLANK                                                      | 2200
      |     |GO TO 30                                                                | 2300
   20 |     |FRSTSB = .FALSE.                                                        | 2400
      |     |CALL EQUAT4( 6,JCARD, KARD )                                            | 2500
      |     |DO 25 I=1,6                                                             | 2600
   25 |     |  PCARD(I) = BLANK                                                      | 2700
   30 |     |J = 7                                                                   | 2800
      |     |IF ( NCHAR .LE. 0 ) GO TO 42                                            | 2900
      |     |DO 40 I=1,NCHAR                                                         | 3000
      |     |  OCHAR = TEXT(I)                                                       | 3100
      |     |  JCARD(J) = ICHAR                                                      | 3200
   40 |     |  J = J+1                                                               | 3300
      |     |JCARD(J) = BLANK                                                        | 3400
      |     |JCARD(J+1) = KSPK(KEY)                                                  | 3500
      |     |JCARD(J+2) = BLANK                                                      | 3600
      |     |J = J+3                                                                 | 3700
C-----|-    |---- MOVE DELETED TEXT TO NEW CARD IMAGE AND BLANK OUT REMAINDER         | 3800
      |     |JMAX  = 72                                                              | 3900
      |     |IF ( L1 .GE. 0 ) GO TO 41                                               | 4000
      |     |JCS = -L1                                                               | 4100
      |     |GO TO 42                                                                | 4200
   41 |     |K = KOL                                                                 | 4300
      |     |KOL = L1                                                                | 4400
      |     |CALL MOVER( 0,DMMY,DMMY,0 )                                             | 4500
      |     |JCS = JC                                                                | 4600
      |     |KOL = K                                                                 | 4700
   42 |     |CALL MOVER( 0,DMMY,DMMY,0 )                                             | 4800
      |     |K = JCS                                                                 | 4900
  420 |     |JCARD(J) = KARD(K)                                                      | 5000
      |     |K = K+1                                                                 | 5100
   43 |     |J = J+1                                                                 | 5200
      |     |IF ( J .LE. JMAX ) GO TO 50                                             | 5300
      |     |CALL PAG( 1 )                                                           | 5400
      |     |WRITE (N4,80) JCARD                                                     | 5500
```

```
      |  |CALL TRANCE( JCARD,JMAX )                                                  5600
      |  |DO 46 L=1,5                                                                5700
   46 |  |  JCARD(L) = BLANK                                                         5800
      |  |J = 7                                                                      5900
      |  |JCARD(J-1) = KSPK(8)                                                       6000
   50 |  |IF ( K .LT. JC ) GO TO 420                                                 6100
      |  |JC = JCS                                                                   6200
      |  |IF ( J .EQ. 7 ) GO TO 100                                                  6300
      |  |DO 70 L=J,72                                                               6400
   70 |  |  JCARD(L) = BLANK                                                         6500
      |  |CALL PAG( 1 )                                                              6600
      |  |WRITE (N4,80) JCARD                                                        6700
      |  |CALL TRANCE( JCARD,JMAX )                                                  6800
   80 |  |FORMAT(8X80A1)                                                             6900
C-----|--|---- MOVE TEXT INTO OLD IMAGE AREA                                         7000
  100 |  |IF ( KEY .GT. 1 ) RETURN                                                   7100
      |  |CALL MOVER( NCHAR,TEXT,1,0 )                                               7200
      |  |RETURN                                                                     7300
      |  |  END                                                                      7400
```

```
      |  |SUBROUTINE MOVER1 ( NT, TEXT, INV, MT)                                  |  100
      |  |INTEGER BLANK                                                           |  200
      |  |LOGICAL OSTMTN, FRSTSB                                                  |  300
      |  |LOGICAL*1 TEXT(1), OCHAR, OSEQQ(8)                                      |  400
      |  |REAL*8 SEQ1, SEQ2, SEQ3                                                 |  500
      |  |COMMON /SEQ/ SEQ1, SEQ2, SEQ3                                           |  600
      |  |COMMON / DELTA/ MOVE, JCARD(80), ICARD(1600), IOL, NCARD                 |  700
     1|  |    , OSTMTN, FRSTSB, IC, JC                                            |  800
      |  |COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K      |  900
     1|  |RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM            | 1000
     2|  |  ,NERR,NERO,MEMKEY                                                     | 1100
      |  |COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                            | 1200
      |  |COMMON /NU/ NCHM                                                        | 1300
      |  |EQUIVALENCE ( OCHAR, ICHAR ), ( OSEQQ(1),SEQ2 )                         | 1400
      |  |DATA BLANK / 1H  /                                                      | 1500
C-----|--|---- TEST FOR CHARACTERS MOVED TO NEW CARD                              | 1600
C     |  |    KOL      COLUMN COUNTER IN DEBLANKED CARD                           | 1700
C     |  |    IC       COLUMN POINTER IN OLD CARD                                 | 1800
C     |  |    JC       COLUMN POINTER IN NEW CARD                                 | 1900
C     |  |    NT = NUMBER OF CHARACTERS FROM THE TEXT FIELD TO BE                 | 2000
C     |  |         MOVED TO THE NEW CARD IMAGE.                                   | 2100
C     |  |  TEXT = CHARACTERS TO BE MOVED                                         | 2200
C     |  |   INV = INCREMENT IN TEXT FIELD ( 1=LOGICAL*1, 4=INTEGER)              | 2300
C     |  |    MT = NUMBER OF NON-BLANK CHARACTERS TO SKIP AFTER THE MOVE          | 2400
      |  |IF ( JC .GT. 1 ) GO TO 10                                               | 2500
C-----|--|---- MOVE STATEMENT NUMBER FIELD                                        | 2600
     1|  |CALL EQUAT4( 6,JCARD, KARD )                                            | 2700
      |  |IC = 7                                                                  | 2800
      |  |JC = 7                                                                  | 2900
      |  |MOVE = 1                                                                | 3000
   10 |  |N = NT                                                                  | 3100
      |  |M = IABS( MT )                                                          | 3200
      |  |NN = KOL-IC+1                                                           | 3300
C     |  |                                                    )518,'01 REVOM '(TAMR| 3400
C     |  |                                                NN,CJ,CI,LOK,TM,TN ,0101 TNI| 3500
      |  |IF ( NN .LE. 0 ) GO TO 30                                               | 3600
C-----|--|---- MOVE KOL-IOL CHARACTERS FROM OLD CARD TO NEW CARD                   | 3700
      |  |IF ( N .LT. 0 ) GO TO 20                                                | 3800
      |  |CALL EQUAT4( NN,JCARD(JC),KARD(IC) )                                    | 3900
      |  |JC = JC + NN                                                            | 4000
   20 |  |IC = KOL + 1                                                            | 4100
   30 |  |IF ( N .LE. 0 ) GO TO 50                                                | 4200
C-----|--|---- MOVE N CHARACTERS FROM TEXT FIELD TO NEW CARD.                     | 4300
      |  |IT = 1                                                                  | 4400
      |  |DO 40 I=1,N                                                             | 4500
      |  |  OCHAR = TEXT(IT)                                                      | 4600
      |  |  IT = IT+INV                                                           | 4700
      |  |  JCARD(JC) = ICHAR                                                     | 4800
      |  |  JC = JC+1                                                             | 4900
   40 |  |  CONTINUE                                                              | 5000
   50 |  |IF ( M .LE. 0 ) GO TO 55                                                | 5100
C-----|--|---- COUNT PAST M NONBLANKS                                             | 5200
      |  |NNB = 0                                                                 | 5300
      |  |DO 52 I=IC,KMAX                                                         | 5400
      |  |  IF ( KARD(I) .EQ. KBL ) GO TO 52                                      | 5500
```

```
      NNB = NNB + 1                                                     5600
      IF ( NNB .GE. M ) GO TO 54                                        5700
   52 CONTINUE                                                          5800
   54 KOL = I                                                           5900
      IC  = I + 1                                                       6000
   55 CONTINUE                                                          6100
C                                                )518,'55 REVOM '(TAMR  6200
C                                              CJ,CI,LOK,TM,TN ,5555 TNI6300
  160 RETURN                                                            6400
      END                                                               6500
```

```
      SUBROUTINE MULTEQ( KEQ )
      DIMENSION LOCEQ(80)
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K
     1RSX
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)
      COMMON / DELTA/ MOVE, JCARD(80)
      KER = KEQ
      LOCEQ(2) = KER
      LOCEQ(1) = 6
      NEQ = 2
   10 KER = KER + 1
      IF ( KARD(KER) .EQ. KBL ) GO TO 10
      IF ( KARD(KER) .GE. ZERO  .AND.  KARD(KER) .LE. NINE ) GO TO 80
      DO 50 I=KER, KMAX
        IF ( KARD(I) .EQ. KSPK(1) ) GO TO 100
        IF ( KARD(I) .EQ. KSPK(10) ) GO TO 80
   50   CONTINUE
   80 CALL SUBSUB
      GO TO 200
C------ MULTIPLE REPLACEMENT STATEMENT
  100 NEQ = NEQ + 1
      KER = I
      LOCEQ(NEQ) = KER
      GO TO 10
C------ WRITE NEW CARDS FROM OLD
  200 CONTINUE
      CALL EQUAT4( 6,JCARD,KARD )
      LAST = KMAX
      IF ( KARD(I) .EQ. KSPK(10) ) LAST = I - 1
      NEQX = NEQ - 1
      J = NEQX
      DO 250 I=1,NEQX
        KK = LOCEQ(J)
        N = LAST - KK
        CALL EQUAT4( N,JCARD(7),KARD(KK+1) )
        CALL TRANCE( JCARD, N+6 )
        JCARD(1) = KBL
        CALL EQUAT4(  5,JCARD(2),JCARD(1) )
        LAST = LOCEQ(J+1) - 1
  250   J = J - 1
      MOVE = -1
      RETURN
        END
```

```
      SUBROUTINE PAG (N)                                                    100
C                                                                           200
C     THIS SUBROUTINE DOES THE GENERAL PAGE COUNTING FOR INDEX.             300
C                                                                           400
      LOGICAL KSBO, KCDO                                                    500
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K    600
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM          700
     2  ,NERR,NERO,MEMKEY                                                  800
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                          900
      COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE,KSBO,KCDO,KOUT(80)            1000
      COMMON /KAPPA/ KAP(11)                                                1100
      COMMON /OMEGA/ KLAST(4),KSTOP(4)                                      1200
      COMMON /SIGMA/ KSTIJ(11,60)                                           1300
      COMMON /TAPES/ N1, N2, N3, N4, N5, N6                                 1400
      K = IABS( N )                                                         1500
      IF ( N .LE. 0 ) KCDO = .FALSE.                                        1600
      IF ( KSBO ) K = K+2                                                   1700
      IF ( KCDO ) K = K+2                                                   1800
      IF ( LINE+K .LE. MXLI ) GO TO 10                                      1900
      WRITE (N6,100)                                                        2000
      LINE = 0                                                              2100
   10 IF ( KSBO ) WRITE (N6,110) JOB                                        2200
      IF ( KCDO ) WRITE (N6,120) KOUT                                       2300
      LINE = LINE+K                                                         2400
      KSBO = .FALSE.                                                        2500
      KCDO = .FALSE.                                                        2600
      RETURN                                                                2700
  100 FORMAT(1H1)                                                           2800
  110 FORMAT(1H0,13X,66A1)                                                  2900
  120 FORMAT(8H0  ---  80A1,5H  ---)                                        3000
      END                                                                   3100
```

```
      |  |SUBROUTINE RETRNI( MODE )                                                  100
      |  |LOGICAL*1 OTAB(256), ODX(4), OCHAR                                         200
      |  |LOGICAL FIRST                                                              300
      |  |COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K         400
     1|  |RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM               500
     2|  |  ,NERR,NERO,MEMKEY                                                        600
      |  |COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                               700
      |  |COMMON / GAMMA/ OTAB                                                       800
      |  |EQUIVALENCE ( ODX(1),I ), ( OCHAR,KRSX )                                   900
      |  |FIRST = .TRUE.                                                            1000
      |  |SEQ3 = SEQ2                                                               1100
      |  |NVRSMT = 0                                                                1200
      |  |I = 0                                                                     1300
      |  |GO TO 11                                                                  1400
    5 |  |KOL = KOL - 1                                                             1500
   11 |  |CALL SYMBOL                                                               1600
      |  |IF ( KODE .LE. 0 ) GO TO 100                                              1700
      |  |IF ( KODE .EQ. MINA ) CALL UPDATE                                         1800
      |  |IF ( .NOT. FIRST ) GO TO 13                                               1900
      |  |FIRST = .FALSE.                                                           2000
      |  |ITYPE = 1                                                                 2100
      |  |IF ( KARD(KOL) .NE. KSPK(8) ) GO TO 13                                    2200
C-----|-|---- AN ASTERISK IN THIS POSITION INDICATES AN EXPLICIT                     2300
C     |  |     TYPE DECLARATION SIZE SPECIFICATION.                                 2400
C     |1|.E., INTEGER FUNCTION CALC*2(  ...  )                                     2500
      |  |LAST = KOL + 3                                                            2600
      |  |I = KOL                                                                   2700
      |  |DO 31 J=KOL,LAST                                                          2800
   30 |  |  IF ( KARD(I+1) .EQ. KSPK(3) ) GO TO 35                                  2900
      |  |  IF ( KARD(I+1) .NE. KBL ) GO TO 31                                      3000
      |  |  I = I + 1                                                               3100
      |  |  GO TO 30                                                                3200
   31 |  |  CONTINUE                                                                3300
      |  |GO TO 41                                                                  3400
C-----|-|---- DROP SIZE SPECIFICATION                                               3500
   35 |  |KOL = KOL - 1                                                             3600
      |  |CALL MOVER( 0,DMMY,DMMY,I-KOL )                                           3700
      |  |KOL = I + 1                                                               3800
   13 |  |CONTINUE                                                                  3900
      |  |IF ( KARD( KOL ) .EQ. KSPK(4) ) KARD(KOL) = KBL                           4000
      |  |IF ( KARD(KOL) .EQ. KBL ) GO TO 41                                        4100
      |  |KRSX = KARD(KOL)                                                          4200
      |  |ODX(4) = OCHAR                                                            4300
      |  |ODX(4) = OTAB(I)                                                          4400
      |  |IF ( I .LE. 36 ) GO TO 5                                                  4500
   41 |  |KOL = KOL+1                                                               4600
      |  |IF ( KOL .GT. KMAX ) RETURN                                               4700
      |  |GO TO 13                                                                  4800
  100 |  |CONTINUE                                                                  4900
      |  |RETURN                                                                    5000
      |  |  END                                                                     5100
```

```
      SUBROUTINE SIXFR7                                                      100
C                                                                            200
C     THIS SUBROUTINE SHORTENS 7 CHARACTER VARIABLE NAMES TO 6               300
C     CHARACTERS AND CHECKS FOR UNIQUENESS THEN ADDS THE NEW                 400
C     NAMES TO THE SYMBOL TABLE AND CROSSREFERENCES OLD AND NEW              500
C     NAMES.                                                                 600
C                                                                            700
      DIMENSION ONM1(16),VOWEL(6),DOLLAR(2)                                  800
      REAL*8 NAME(2),SYMTAB(2,800),NEWNM(800)                                900
      INTEGER*2 IREF(800)                                                   1000
      LOGICAL*1 ONM(16)                                                     1100
      LOGICAL VAXFLG,DGFLAG,CHGFLG                                          1200
C                                                                           1300
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K    1400
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM          1500
     2  ,NERR,NERO,MEMKEY                                                  1600
      COMMON /HASH/ ILOC, LOC, NAME                                         1700
      COMMON /SIX7/ NEWNM,IREF                                              1800
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                          1900
      COMMON /FLAGS/ DGFLAG,VAXFLG                                          2000
C                                                                           2100
      EQUIVALENCE (NAME(1),ONM(1)),(ONM(1),ONM1(1))                         2200
      EQUIVALENCE (SYMTAB(1,1),LIST(4192))                                  2300
C                                                                           2400
      DATA VOWEL /1HA,1HE,1HI,1HO,1HU,1HY/                                  2500
      DATA CHGFLG /.FALSE./                                                 2600
      DATA DOLLAR /1H$,1H_/                                                 2700
      DO 5 I=1,800                                                          2800
        IREF(I)=0                                                           2900
    5   CONTINUE                                                            3000
      N4=1                                                                  3100
C                                                                           3200
C     FIND 6 OR LESS CHARACTER NAMES IN SYMTAB AND MOVE TO NEWNM            3300
C                                                                           3400
      DO 200 I=1,ILOC                                                       3500
        NAME(1)=SYMTAB(1,I)                                                 3600
        IF(ONM1(I).NE.KBL)GO TO 200                                         3700
        NEWNM(I)=NAME(1)                                                    3800
  200   CONTINUE                                                            3900
C                                                                           4000
C     SEARCH FOR NAMES TO BE CHANGED                                        4100
C                                                                           4200
    7 N7=2                                                                  4300
   10 N6=N7                                                                 4400
      N1=N4                                                                 4500
      DO 20 I=N1,ILOC                                                       4600
        IF (.NOT.VAXFLG)GO TO 15                                            4700
        IF(CHGFLG)GO TO 14                                                  4800
        NAME(1)=SYMTAB(1,I)                                                 4900
        NAME(2)=SYMTAB(2,I)                                                 5000
   14   DO 240 J=N6,16                                                      5100
          IF(ONM1(J).EQ.KBL)GO TO 16                                        5200
          IF(ONM1(J).NE.DOLLAR(1).AND.ONM1(J).NE.DOLLAR(2))GO TO 240        5300
          CHGFLG=.TRUE.                                                     5400
          J1=J                                                              5500
```

```
            N4=I
            N7=J
            GO TO 50
 240        CONTINUE
          GO TO 16
 15       NAME(1)=SYMTAB(1,I)
          NAME(2)=SYMTAB(2,I)
 16       IF(ONM1(8).EQ.KBL)GO TO 17
          CHGFLG=.FALSE.
          N4=I
          N2=1
          GO TO 30
 17       IF(.NOT.CHGFLG)GO TO 20
          CHGFLG=.FALSE.
          J1=8
          N4=I
          N2=1
          GO TO 75
 20       CONTINUE
 25     RETURN
C
C       LOOK FOR LAST VOWEL IN NAME
C
 30     DO 40 J=N2,14
          IF(ONM1(17-J).EQ.KBL)GO TO 40
          DO 40 K=1,6
            IF(ONM1(17-J).NE.VOWEL(K))GO TO 40
            J1=17-J
            GO TO 50
 40         CONTINUE
        GO TO 90
C
C       VOWEL FOUND - DELETE AND CREATE NEW NAME
C
 50     IF(J1.EQ.16)GO TO 70
        DO 60 J=J1,15
          ONM1(J)=ONM1(J+1)
          IF(ONM1(J+1).EQ.KBL)GO TO 71
 60       CONTINUE
 70     ONM1(16)=KBL
 71     IF(CHGFLG)GO TO 10
        IF(ONM1(8).EQ.KBL)GO TO 75
        IF(J1.EQ.3)GO TO 90
        N2=18-J1
        GO TO 30
C
C       LOOK FOR DUPLICATE
C
 75     DO 80 I=1,ILOC
          IF(NEWNM(I).NE.NAME(1))GO TO 80
          IF(J1.EQ.3)GO TO 90
          IF(J1.EQ.2)GO TO 100
          IF(J1.EQ.1)GO TO 115
          N2=18-J1
          GO TO 30
```

```
   80    CONTINUE
C
C     NAME IS UNIQUE - ADD NAME TO NEWNM AND FLAG IREF WITH THE
C     NUMBER OF CHARACTERS IN THE NEW NAME
C
      NEWNM(N4)=NAME(1)
      DO 85 I=3,8
        IF(ONM1(I).NE.KBL)GO TO 85
        IREF(N4)=I-2
        GO TO 86
   85   CONTINUE
   86 N4=N4+1
      IF(N4.LE.ILOC)GO TO 7
      GO TO 25
C
C     NO MORE VOWELS - DELETE LAST CHARACTER
C
   90 J1=2
      N3=0
      IF(ONM1(8).EQ.KBL)GO TO 97
      DO 95 I=8,16
        ONM1(I)=KBL
   95   CONTINUE
      ISAVE=7
      GO TO 75
   97 DO 98 I=3,7
        IF(ONM1(10-I).EQ.KBL)GO TO 98
        ISAVE=10-I
        GO TO 100
   98   CONTINUE
      ISAVE=3
C
C     STILL NOT UNIQUE, CHANGE LAST CHARACTER TO NUMBER
C
  100 IF(N3.EQ.10)GO TO 110
      N3=N3+1
      ONM1(ISAVE)=KDIG(N3)
      GO TO 75
C
C     STILL NOT UNIQUE, CHANGE LAST CHARACTER TO LETTER
C
  110 N5=0
      J1 = 1
  115 IF(N5.EQ.26)GO TO 120
      N5=N5+1
      ONM1(ISAVE)=KABC(N5)
      GO TO 75
  120 NAME(1)=SYMTAB(1,N4)
      NAME(2)=SYMTAB(2,N4)
      WRITE(6,140) NAME
      N4=N4+1
      IF(N4.LE.ILOC)GO TO 7
      GO TO 25
  140 FORMAT(27HOCAN'T FIND UNIQUE NAME FOR ,A8,A8)
        END
```

PAGE 3 OF SIXFR7

```
      SUBROUTINE SUB6(KOL1,KOL2)                                        100
C                                                                       200
C     THIS SUBROUTINE SUBSTITUTES THE 6 CHARACTER NAME FOR THE 7        300
C     CHARACTER NAME.                                                   400
C                                                                       500
      LOGICAL DGFLAG,VAXFLG                                             600
      LOGICAL*1 ONM(8),ONS                                              700
      COMPLEX*16 SYMTAB(800)                                            800
      REAL*8 NEWNM(800),NME                                             900
      INTEGER*2 IREF(800)                                              1000
C                                                                      1100
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K 1200
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM      1300
     2  ,NERR,NERO,MEMKEY                                              1400
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                     1500
      COMMON /HASH/ ILOC, LOC, NAME                                    1600
      COMMON /FLAGS/ DGFLAG,VAXFLG                                     1700
      COMMON /SIX7/ NEWNM,IREF                                         1800
C                                                                      1900
      EQUIVALENCE (ONM(1),NME)                                         2000
      EQUIVALENCE(ONS,KSAVE)                                           2100
      EQUIVALENCE (SYMTAB(1),LIST(5192))                               2200
C                                                                      2300
      KEND=KMAX                                                        2400
      ITYPE=1                                                          2500
      KOL=KOL1                                                         2600
      KMAX=KOL2                                                        2700
      CALL SYMBOL                                                      2800
      CALL UPDATE                                                      2900
      IF(IREF(LOC).EQ.0)GO TO 70                                       3000
      NME=NEWNM(LOC)                                                   3100
      K=KOL2-KOL1                                                      3200
      L=K+1                                                            3300
      DEL=L-IREF(LOC)                                                  3400
      IF(DEL.EQ.0.OR.DEL.EQ.1)GO TO 30                                 3500
      IF(KOL2.EQ.KEND)GO TO 30                                         3600
      IF(DEL.LT.0)GO TO 20                                             3700
      DO 15 I=KOL2,KEND                                                3800
        KARD(I-DEL)=KARD(I)                                            3900
   15   CONTINUE                                                       4000
      KOL2=KOL2-DEL                                                    4100
      KEND=KEND-DEL                                                    4200
      GO TO 30                                                         4300
   20 DO 25 I=KOL2,KEND                                                4400
        COUNT=KEND+KOL2-I                                              4500
        KARD(COUNT-DEL)=KARD(COUNT)                                    4600
   25   CONTINUE                                                       4700
      KOL2=KOL2-DEL                                                    4800
      KEND=KEND-DEL                                                    4900
   30 DO 40 I=KOL1,KOL2                                                5000
        ONS=ONM(I+2-KOL1)                                              5100
        KARD(I)=KSAVE                                                  5200
   40   CONTINUE                                                       5300
      KMAX=KEND                                                        5400
   60 IF(.NOT.DGFLAG)GO TO 70                                          5500
```

```
      WRITE(6,80)SYMTAB(LOC),NEWNM(LOC)                                 5600
   70 RETURN                                                            5700
   80 FORMAT(1X,A16,13H REPLACED BY ,A8,59H IN SUBROUTINE, ENTRY, FUNCTI5800
     C ON, OR NAMED COMMON STATEMENT.)                                  5900
        END                                                             6000
```

```
      SUBROUTINE SLASHS( LAST )
      LOGICAL*1 OCHAR, ONAME(8)
      INTEGER IDATA(4)
      REAL*8 NAME(2), NAMER(3)
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM
     2 ,NERR,NERO,MEMKEY
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)
      COMMON / DELTA/ MOVE, JCARD(80), ICARD(1600), IOL, NCARD
     1  , OSTMTN, FRSTSB, IC, JC
      COMMON /HASH/ ILOC,LOC,NAME
      EQUIVALENCE ( IDATA(1), NAMER(1) )
      DATA NAMER / 8H    DATA, 8H          /
      M = LAST-KOL
      IF ( M .LE. 0 ) GO TO 11
      CALL MOVER( 0,DUMMY,DUMMY,M )
      KOL = LAST
   11 CALL SYMBOL
C                                       )1A7X2,1AX2,5I5,'11 SHSALS '(TAMR
C      )6LOK,LOK=XPI,)XPI(DRAK (,XSRK,COL,XAMK,LOK,TNCPL ,RAPI ,116 TNI
C                                                             6+LOK = 6L
      IF ( KOL .GT. KMAX ) GO TO 100
      IF ( KODE .LT. MINA  .OR.  KODE .GT. MAXA ) GO TO 12
      NAMER(2) = NAME(1)
      NAMER(3) = NAME(2)
      CALL UPDATE
      IF ( KRSX .NE. KSPK(3) ) GO TO 13
C-----------  SYMBOLS WITHIN PARENTHESES ARE SKIPPED.
      IF ( KOL .GT. KMAX ) GO TO 21
   12 KOL = KOL+1
      KRSX = KARD(KOL)
      IF ( KRSX .NE. KSPK(5) ) GO TO 12
 1200 KOL = KOL + 1
      KRSX = KARD(KOL)
      IF ( KRSX .EQ. KBL ) GO TO 1200
C-----------  TEST FOR DATA IN TYPE DEFINITION STATEMENTS
   13 IF ( KRSX .NE. KSPK(4) ) GO TO 11
      L1 = KOL-1
      KOL = L1
C                                               )5I5,'31 SHSALS '(TAMR
C                                                 CJ ,CI ,LOK ,316 TNI
      CALL MOVER( 0,DMMY,DMMY,1 )
      JCS = JC
   14 CALL SYMBOL
C                                       )1A7X2,1AX2,5I5,'41 SHSALS '(TAMR
C      )6LOK,LOK=XPI,)XPI(DRAK (,XSRK       ,SCJ,EDOK,CJ,CI,LOK,416 TNI
C                                                             6+LOK = 6L
      IF ( KODE .LT. 0 ) GO TO 21
      IF ( KRSX .NE. KSPK(4) ) GO TO 14
      IF ( JC .GT. JCS ) L1 = -JCS
      CALL INSRTR( L1,IDATA(2),12,4 )
      KOL = KOL+1
      GO TO 11
   21 CALL ERROR
  100 CONTINUE
```

```
      RETURN                                                    5600
        END                                                     5700
```

```
      SUBROUTINE SORT( IGO )
      REAL*8 NAME(2), SYMTAB(2,800), BLANK, TDNAME(4), OUTBF(10), STAR
      REAL*8 SEQ1, SEQ2, SEQ3
      LOGICAL*1 SYML1(8,1)
      LOGICAL OUPDAT
      LOGICAL*1 MORE, ONAME(8), ODX(4), OIDX, OTAB(256), OCOMMA, OBLANK
      LOGICAL TYTL / .FALSE. /
     1 , OOBF(1)
      DIMENSION JCDICT(27), NX(5), JLOC(1)
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLFAR(8),KMAX,KODE,KOL,K
     1RSX,LIST(8191)
      COMMON /SEQ/ SEQ1, SEQ2, SEQ3
      COMMON /HASH/ ILOC, LOC, NAME
      COMMON /XUPDAT/ OUPDAT, ICDICT(27)
      COMMON /GAMMA/ OTAB
      EQUIVALENCE ( ONAME(1),NAME ), ( ODX(1),IDX ), ( ODX(4), OIDX ),
     *  ( OCOMMA,COMMA ), ( OBLANK,BLANK ), ( OOBF(1),OUTBF(1) )
      EQUIVALENCE ( SYMTAB (1),LIST(4192) ), ( JLOC(1),LIST(7392) )
      EQUIVALENCE ( SYMTAB(1),SYML1(1,1) )
      DATA TDNAME / 8HCOMPLEX , 8HINTEGER , 8HLOGICAL , 8HREAL    /
      DATA JCDICT / 8*5, 6*3, 12*5, 0 /, COMMA, BLANK / 1H,, 1H  /
      DATA STAR / 8H******** /
      IF ( IGO .EQ. 0 ) GO TO 200
      TYTL = .FALSE.
C
      IF ( ILOC .LE. 0 ) RETURN
      DO 50 I=1,ILOC
        NAME(1) = SYMTAB(1,I)
        IF ( ONAME(8) .EQ. OBLANK ) GO TO 50
        IF ( TYTL ) GO TO 40
        TYTL = .TRUE.
        PRINT 70, JOB
   40   PRINT 80, NAME(1)
   50   CONTINUE
  200 RETURN
   70 FORMAT('0ILLEGAL VARIABLE NAMES IN ',66A1/)
   80 FORMAT(2XA8)
        END
```

PAGE 1 OF SORT

```
      SUBROUTINE STARL (K)                                              100
C-----SUBROUTINE STARL FILLS (K[OFF) WITH (=LAST).                      200
      DIMENSION K(80), L(80)                                            300
      DATA  L / 1H=, 1HL, 1HA, 1HS, 1HT, 75=1H  /                       400
      DO 9 I=1,80                                                       500
    9   K(I) = L(I)                                                     600
      RETURN                                                            700
        END                                                             800
```

```
        SUBROUTINE SUBSUB
        REAL*8 SSNAME(25)
        INTEGER AMPSND, L1(25), KLASS(25)
        COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K
      1 RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM
      2  ,NERR,NERO,MEMKEY
        COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)
        COMMON /DELTA/ MOVE, JCARD(80), ICARD(1600), IOL, NCARD
      1    , OSTMTN, FRSTSB, IC, JC, IPAR, L1, KLASS
        COMMON /NU/ NH, KOLSX
        COMMON /HASH/ ILOC, LOC
        DATA AMPSND / 1H& /
        DATA SSNAME /
      1     8HI5B5B17 , 8HI5B5B27 , 8HI5B5B37 , 8HI5B5B47 , 8HI5B5B57 ,
      2     8HI5B5B67 , 8HI5B5B77 , 8HI5B5B87 , 8HI5B5B97 , 8HI5B5BA7 ,
      3     8HI5B5BB7 , 8HI5B5BC7 , 8HI5B5BD7 , 8HI5B5BE7 , 8HI5B5BF7 ,
      4     8HI5B5BG7 , 8HI5B5BH7 , 8HI5B5BI7 , 8HI5B5BJ7 , 8HI5B5BK7 ,
      5     8HI5B5BL7 , 8HI5B5BM7 , 8HI5B5BN7 , 8HI5B5BO7 , 8HI5B5BP7   /
C----- ---- LOOK FOR SUBSCRIPTED SUBSCRIPTS AND CORRECT THEM IF FOUND.
        IPAR = 1
        LPCNT = 0
   10   IF ( KARD(KOL+1) .EQ. AMPSND ) GO TO 40
   11   CALL SYMBOL
        KOL6 = KOL+6
        PRINT 611, IPAR, LPCNT,KOL,KMAX,LOC,KRSX,( KARD(IPX),IPX=KOL,KOL6)
  611   FORMAT(' SUBSUB 11',5I5,2XA1,2X7A1)
        IF ( KOL .GT. KMAX ) GO TO 100
        IF ( KRSX .EQ. KSPK(10) ) GO TO 100
        L1(IPAR) = KOLSX
        CALL UPDATE
        IF ( KRSX .NE. KSPK(3) ) GO TO 20
C----- ---- ARRAY NAME OR FUNCTION REFERENCE
        KLASS(IPAR) = LOC
        ILPG = IPAR
        IPAR = IPAR+1
        IF ( IPAR .GT. 25 ) GO TO 200
        IF ( LOC .GE. 0 ) GO TO 15
C----- ---- ARRAY
   13   CALL MOVER( 0,DMMY,DMMY,0 )
        L1(IPAR-1) = -JC
        PRINT 613, IPAR, JC
  613   FORMAT(' SUBSUB 13',6I5)
        GO TO 10
C----- ---- FUNCTION REFERENCE
   15   K = KOL
        KOL = L1(IPAR-1)
        CALL FNCHEK( 1 )
        PRINT 615, IPAR, K, KOL
  615   FORMAT(' SUBSUB 15',6I5)
        KOL = K
        GO TO 10
   20   CONTINUE
        PRINT 620
  620   FORMAT(' SUBSUB 20',6I5)
        IF ( KRSX .EQ. KSPK(5) ) GO TO 25
```

```
      |  |IF ( KRSX .NE. KSPK(2) ) GO TO 10                                                                    |  5600
C-----|- |---- COMMA.  BRANCH IF INSIDE FUNCTION                                                               |  5700
   22 |  |CONTINUE                                                                                             |  5800
      |  |PRINT 622, IPAR, L1(IPAR-1),ILPG,LPCNT,KOL                                                           |  5900
  622 |  |FORMAT(' SUBSUB 22',6I5)                                                                             |  6000
      |  |IF ( IPAR .LE. 1 ) GO TO 10                                                                          |  6100
      |  |IF ( L1(IPAR-1) .GE. 0 ) GO TO 10                                                                    |  6200
      |  |IF ( IPAR .GT. ILPG ) GO TO 13                                                                       |  6300
      |  |ILPG = ILPG-1                                                                                        |  6400
      |  |LPCNT = LPCNT+1                                                                                      |  6500
      |  |KOL = KOL-1                                                                                          |  6600
      |  |CALL INSRTR( L1(IPAR-1),SSNAME(LPCNT),7,1 )                                                          |  6700
      |  |KOL = KOL+1                                                                                          |  6800
      |  |GO TO 13                                                                                             |  6900
C-----|- |---- RIGHT PAREN.  IF FIRST LEVEL, SKIP.                                                             |  7000
   25 |  |CONTINUE                                                                                             |  7100
      |  |IPAR = IPAR-1                                                                                        |  7200
      |  |PRINT 625, IPAR, L1(IPAR), ILPG, LPCNT, KOL                                                          |  7300
  625 |  |FORMAT(' SUBSUB 25',6I5)                                                                             |  7400
      |  |IF ( L1(IPAR) .GE. 0 ) GO TO 30                                                                      |  7500
   28 |  |IF ( IPAR .GE. ILPG ) GO TO 30                                                                       |  7600
      |  |ILPG = ILPG-1                                                                                        |  7700
      |  |LPCNT = LPCNT+1                                                                                      |  7800
      |  |KOL = KOL-1                                                                                          |  7900
      |  |CALL INSRTR( L1(IPAR),SSNAME(LPCNT),7,1 )                                                            |  8000
      |  |KOL = KOL+1                                                                                          |  8100
   30 |  |KOL = KOL+1                                                                                          |  8200
      |  |IF ( KOL .GT. KMAX ) GO TO 100                                                                       |  8300
      |  |KRSX = KARD(KOL)                                                                                     |  8400
      |  |IF (   KRSX     .EQ. KSPK(5) ) GO TO 25                                                              |  8500
      |  |IF (   KRSX     .EQ. KSPK(2) ) GO TO 22                                                              |  8600
      |  |IF (   KRSX     .EQ. KBL ) GO TO 30                                                                  |  8700
      |  |IF (   KRSX     .EQ. KSPK(10) ) GO TO 100                                                            |  8800
      |  |GO TO 10                                                                                             |  8900
   40 |  |CONTINUE                                                                                             |  9000
      |  |IF ( KOL .GE. KMAX ) RETURN                                                                          |  9100
C-----|- |---- LOOK FOR &XX WHERE XX IS A STATEMENT NUMBER                                                      |  9200
C-----|- |---- NOT A STATEMEMT NUMBER UNLESS PRECEEDED BY A ( OR ,                                             |  9300
      |  |IF ( KARD(KOL).NE.KSPK(2) .AND. KARD(KOL).NE.KSPK(3) ) GO TO 11                                      |  9400
      |  |KOL = KOL+1                                                                                          |  9500
      |  |GO TO 10                                                                                             |  9600
  100 |  |CONTINUE                                                                                             |  9700
      |  |RETURN                                                                                               |  9800
  200 |  |CONTINUE                                                                                             |  9900
      |  |CALL ERR( 17,DMMY,DMMY )                                                                             | 10000
      |  |WRITE (6,222) KARD                                                                                   | 10100
  222 |  |FORMAT(10X80A1)                                                                                      | 10200
      |  |STOP                                                                                                 | 10300
      |  |  END                                                                                                | 10400
      |  |                                                                                                     |
      |  |                                                                                                     |
      |  |                                                                                                     |
      |  |                                                                                                     |
      |  |                                                                                                     |
```

```
      |  |SUBROUTINE SYMBOL                                                                  |  100
C     |  |                                                                                   |  200
C     |  |THIS SUBROUTINE INSPECTS KARD STARTING AT KOL+1 FOR THE PRESENCE                   |  300
C     |  |OF FORTRAN VARIABLES OR STATEMENT NUMBERS.  IF FOUND, THE SYMBOL                   |  400
C     |  |IS PACKED IN KODE AS A NUMBER WHOSE BASE IS 37.                                    |  500
C     |  |                                                                                   |  600
      |  |LOGICAL*1 ONM(8), ONAME(7), OCHAR, OTAB(256), ODX(4), OCHAL                        |  700
      |  |LOGICAL DTOE, NUMLY                                                                |  800
      |  |REAL*8 NAME, BLANK                                                                 |  900
      |  |COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K                 | 1000
     1|  |RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM                      | 1100
     2|  |  ,NERR,NERO,MEMKEY                                                                | 1200
      |  |COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                                       | 1300
      |  |COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE,KSBO,KCDO                                  | 1400
      |  |COMMON /GAMMA / OTAB                                                               | 1500
      |  |COMMON /DELTA/ MOVE,JCARD(80), ICARD(1600), IOL, NCARD                             | 1600
     1|  |  , OSTMTN, FRSTSB, IC, JC, IPAR, L1(25), KLASS(25)                                | 1700
      |  |COMMON /KAPPA/ KAP(11)                                                             | 1800
      |  |COMMON /NU/ NH, KOLSX                                                              | 1900
      |  |COMMON /SIGMA/ KSTIJ(11,60)                                                        | 2000
      |  |COMMON /OMEGA/ KLAST(4),KSTOP(4)                                                   | 2100
      |  |COMMON /TAPES/ N1, N2, N3, N4, N5, N6                                              | 2200
      |  |COMMON /HASH/ ILOC, LOC, NAME(2), IVAL, NUMLY                                      | 2300
C     |  |                                                                                   | 2400
      |  |DIMENSION KSYM(37)                                                                 | 2500
      |  |EQUIVALENCE (KBL,KSYM(1))                                                          | 2600
      |  |EQUIVALENCE ( ONM(1),NAME(1)),(ONM(2),ONAME(1) ), ( OCHAR,KRSX ),                  | 2700
     *|  |  ( ODX(1), I ), ( OCHAL,KLSX )                                                    | 2800
      |  |DATA BLANK  / 1H  /                                                                | 2900
C     |  |                                                                                   | 3000
C     |  |A    B    C    D    E    F    G    H    I    J    K    L    M                    | 3100
C     |  |1    2    3    4    5    6    7    8    9    10   11   12   13                   | 3200
C     |  |                                                                                   | 3300
C     |  |N    O    P    Q    R    S    T    U    V    W    X    Y    Z                    | 3400
C     |  |14   15   16   17   18   19   20   21   22   23   24   25   26                   | 3500
C     |  |                                                                                   | 3600
C     |  |=    ,    (    /    )    +    -    *    .    $    $/4                           | 3700
C     |  |1    2    3    4    5    6    7    8    9    10   11                             | 3800
C     |  |                                                                                   | 3900
C     |  |START.  SET UP INITIAL CONDITIONS.                                                 | 4000
C     |  |                                                                                   | 4100
1     |  |KODE=0                                                                             | 4200
      |  |NCHAR=0                                                                            | 4300
      |  |I = 0                                                                              | 4400
      |  |NAME(1) = BLANK                                                                    | 4500
      |  |NAME(2) = BLANK                                                                    | 4600
C     |  |                                                                                   | 4700
C     |  |SEARCH FOR FIRST SYMBOL.                                                           | 4800
C     |  |                                                                                   | 4900
2     |  |KLSX=KARD(KOL)                                                                     | 5000
      |  |KOLSX = KOL                                                                        | 5100
    3 |  |KOL=KOL+1                                                                          | 5200
      |  |IF (KOL.GT.KMAX) GO TO 23                                                          | 5300
      |  |KRSX = KARD(KOL)                                                                   | 5400
      |  |ODX(4) = OCHAR                                                                     | 5500
```

```
      |  |ODX(4) = OTAB(I)                                                    5600
      |  |IF ( I - 37 ) 4, 3, 3000                                            5700
C-----|--|---- BRANCH IF APOSTROPHE                                           5800
 3000 |  |IF ( KRSX .EQ. KSPK(11) ) GO TO 24                                  5900
      |  |IF ( KRSX .EQ. KSPK(10) ) GO TO 230                                 6000
      |  |IF ( KRSX .NE. KSPK(3) ) GO TO 2                                    6100
      |  |L1(IPAR) = 0                                                        6200
      |  |KLASS(IPAR) = 0                                                     6300
      |  |IPAR = IPAR+1                                                       6400
      |  |L1(IPAR) = KOL                                                      6500
      |  |GO TO 2                                                             6600
C     |  |                                                                    6700
C     |  |TEST IF FIRST CHARACTER IS NUMERIC.                                 6800
C     |  |                                                                    6900
4     |  |IF ( I .GE. 27 ) GO TO 8                                            7000
C     |  |                                                                    7100
C     |  |FIRST CHARACTER IS ALPHA.  TEST FOR LEADING PERIOD.                 7200
C     |  |IF FIRST ALPHA SYMBOL IS Z, CHECK FOR HEX.  CONSTANT.               7300
C     |  |                                                                    7400
      |  |IF (KLSX.EQ.KSPK(9)) GO TO 18                                       7500
      |  |IF (I.EQ.26) GO TO 19                                               7600
C     |  |                                                                    7700
C     |  |START ALPHA SYMBOL ASSEMBLY.                                        7800
C     |  |                                                                    7900
      |  |KODE = MINA                                                         8000
5     |  |NCHAR=NCHAR+1                                                       8100
      |  |ONAME(NCHAR) = OCHAR                                                8200
      |  |IF (NCHAR.GT.MXCH) GO TO 15                                         8300
    6 |  |KOL=KOL+1                                                           8400
      |  |IF (KOL.GT.KMAX) GO TO 7                                            8500
      |  |KRSX=KARD(KOL)                                                      8600
      |  |ODX(4) = OCHAR                                                      8700
      |  |ODX(4) = OTAB(I)                                                    8800
      |  |IF ( I - 37 ) 5, 6, 7                                               8900
C     |  |                                                                    9000
C     |  |PACK OUT KODE WITH ANY NECESSARY BLANKS AND EXIT.                   9100
C     |  |                                                                    9200
    7 |  |CONTINUE                                                            9300
      |  |RETURN                                                              9400
C     |  |                                                                    9500
C     |  |START NUMBER ASSEMBLY.                                              9600
C     |  |                                                                    9700
    8 |  |CONTINUE                                                            9800
      |  |IVAL = 0                                                            9900
C-----|--|---- SKIP LEADING ZEROES                                            10000
9     |  |IF (I.EQ.27.AND.NCHAR.EQ.0) GO TO 10                                10100
      |  |IVAL = 10*IVAL + I - 27                                             10200
      |  |NCHAR=NCHAR+1                                                       10300
10    |  |KOL=KOL+1                                                           10400
      |  |IF (KOL.GT.KMAX) RETURN                                             10500
      |  |KRSX=KARD(KOL)                                                      10600
      |  |ODX(4) = OCHAR                                                      10700
      |  |ODX(4) = OTAB(I)                                                    10800
      |  |IF ( I .GE. 27 .AND. I .LE. 36 ) GO TO 9                            10900
      |  |IF (KRSX.EQ.KSPK(9)) GO TO 11                                       11000
```

```
      IF ( KRSX .EQ. KBL ) GO TO 10                                     11100
      IF ( NUMLY ) RETURN                                               11200
      IF ( KRSX .EQ. KABC(4)  .OR.  KRSX .EQ. KABC(5) ) GO TO 16        11300
      IF ( KRSX .EQ. KABC(8) ) GO TO 160                                11400
      IF ( KRSX .EQ. KSPK(10) ) GO TO 230                               11500
      RETURN                                                            11600
C---- ---- DECIMAL POINT FOLLOWING NUMBER.  TEST FOR NUMBER             11700
C          CONTINUATION OR LOGICAL OPERATOR.                            11800
   11 IF ( KOL .GE. KMAX ) RETURN                                       11900
 1100 KRSX = KARD(KOL+1)                                                12000
      ODX(4) = OCHAR                                                    12100
      ODX(4) = OTAB(I)                                                  12200
      IF ( I - 37 ) 1105, 1110, 28                                      12300
 1105 IF ( I .GE. 27 ) GO TO 10                                         12400
      KOL = KOL+1                                                       12500
      IF ( I .GE. 16 ) GO TO 15                                         12600
C               A        D  E     G              L     N  O            12700
      GO TO ( 18,15,15,16,12,15,18,15,15,15,15,18,15,18,18 ), I         12800
 1110 KOL = KOL + 1                                                     12900
      GO TO 11                                                          13000
C---- ---- .E FOUND.  TEST FOR NUMBER OR .EQ.                          13100
   12 IF ( KARD(KOL+1) .EQ. KABC(17) ) GO TO 18                         13200
      IF ( KARD(KOL+1) .NE. KBL ) GO TO 10                              13300
      KOL = KOL + 1                                                     13400
      GO TO 12                                                          13500
C                                                                       13600
C     INVALID SYMBOL.  SCAN TO NEXT SPECIAL CHARACTER.                  13700
C                                                                       13800
15    CALL ERROR                                                        13900
      IVAL = 1328                                                       14000
C                                                                       14100
C---- ---- HOLLERITH FIELD.  SUPPRESS D TO E CONVERSION.               14200
160   DTOE = .FALSE.                                                    14300
      KMX = MIN0( KOL+IVAL, KMAX )                                      14400
      GO TO 165                                                         14500
C---- ---- NUMBER.                                                     14600
16    DTOE = .TRUE.                                                     14700
      KMX = KMAX                                                        14800
  165 IF ( DTOE  .AND.  KRSX .EQ. KABC(4) ) KARD(KOL) = KABC(5)         14900
      KOL=KOL+1                                                         15000
      KODE = 1                                                          15100
      IF ( KOL .GT. KMX ) GO TO 23                                      15200
      KRSX = KARD(KOL)                                                  15300
      DO 17 I=1,9                                                       15400
        IF ( KRSX .EQ. KSPK(I) ) RETURN                                 15500
17    CONTINUE                                                          15600
      IF ( KRSX .EQ. KSPK(10) ) GO TO 230                               15700
      GO TO 165                                                         15800
C                                                                       15900
C     SKIP OVER LOGICAL OPERATORS OR CONSTANTS.                         16000
C                                                                       16100
18    KOL=KOL+1                                                         16200
      IF (KOL.GT.KMAX) RETURN                                           16300
      IF (KARD(KOL).NE.KSPK(9)) GO TO 18                                16400
      GO TO 1                                                           16500
```

```
C                                                                       16600
C     FIRST SYMBOL IS Z.  SCAN LINE FOR POSSIBLE HEX. CONSTANT.         16700
C                                                                       16800
C-----  ---- TEST FOR DATA STATEMENT                                    16900
   19   CONTINUE                                                        17000
        IF ( ITYPE .EQ. 7 ) GO TO 190                                   17100
        IF ( KARD(KOL+1) .NE. KSPK(11) ) GO TO 5                        17200
  190   IVAL = 0                                                        17300
        KB1 = KOL+1                                                     17400
        DO 21 IB=KB1,KMAX                                               17500
          DO 20 J=1,10                                                  17600
            IF (KARD(IB).NE.KDIG(J)) GO TO 20                           17700
            IVAL = 16*IVAL + J - 1                                      17800
            GO TO 21                                                    17900
20          CONTINUE                                                    18000
          DO 200 J=1,6                                                  18100
            IF ( KARD(IB) .NE. KABC(J) ) GO TO 200                      18200
            IVAL = 16*IVAL + J + 9                                      18300
            GO TO 21                                                    18400
  200       CONTINUE                                                    18500
          IF ( KARD(IB) .NE. KBL ) GO TO 22                             18600
21        CONTINUE                                                      18700
        GO TO 23                                                        18800
   22   KB2 = IB-1                                                      18900
        IB=KARD(IB)                                                     19000
        IF (IB.EQ.KSPK(2).OR.IB.EQ.KSPK(4).OR.IB.EQ.KSPK(5).OR.IB.EQ.KSPK(  19100
       1 6).OR.IB.EQ.KSPK(7).OR.IB.EQ.KSPK(8).OR.IB.EQ.KSPK(11)) GO TO 220  19200
        GO TO 5                                                         19300
  220   IF ( KB2 .GE. KB1 ) CALL ERR( 15,DMMY,DMMY )                    19400
        GO TO 1                                                         19500
C                                                                       19600
C     END OF LINE EXIT.                                                 19700
C                                                                       19800
   23   KRSX=KBL                                                        19900
230     KODE=-1                                                         20000
        NCHAR=0                                                         20100
        KLSX=KBL                                                        20200
        RETURN                                                          20300
C-----  ---- TEXT BETWEEN APOSTROPHES.  LEAVE SPACE FOR NNNH.  LOOK FOR 20400
C            TRAILING APOSTROPHE AND MOVE TEXT TO JCARD FIELD.          20500
   24   KOL = KOL - 1                                                   20600
        CALL MOVER( 4,4H   H,1,1 )                                      20700
        KH = JC - 1                                                     20800
C-----  ---- LOOP UNTIL TRAILING APOSTROPHE FOUND                       20900
   25   KOL = KOL + 1                                                   21000
        IF ( KOL .GT. KMAX ) GO TO 29                                   21100
        IF ( KARD(KOL) .NE. KSPK(11) ) GO TO 25                         21200
        IF ( KARD(KOL+1) .NE. KSPK(11) ) GO TO 26                       21300
        CALL MOVER( 0,DUMMY,DUMMY,-1 )                                  21400
        GO TO 25                                                        21500
   26   KOL = KOL - 1                                                   21600
        CALL MOVER( 0,DUMMY,DUMMY,-1 )                                  21700
        NH = JC - KH - 1                                                21800
        KODE = 1                                                        21900
        DO 27 J=1,3                                                     22000
```

```
      K1 = NH/10                                                        22100
      K2 = NH - 10*K1                                                   22200
      KH = KH - 1                                                       22300
      JCARD(KH) = KDIG(K2+1)                                            22400
      IF ( K1 .LE. 0 ) GO TO 28                                         22500
   27 NH = K1                                                           22600
   28 KOL = KOL + 1                                                     22700
      KRSX = KARD(KOL)                                                  22800
      IF ( KRSX .EQ. KBL ) GO TO 28                                     22900
      RETURN                                                            23000
C---- ---- IMPROPER END OF APOSTROPHE STRING.                          23100
   29 WRITE (6,30) KARD(KOL-1), KSPK(11)                                23200
   30 FORMAT('0SECOND APOSTROPHE NOT FOUND.',2X2A5)                     23300
      GO TO 28                                                          23400
      END                                                               23500
```

```
      SUBROUTINE TDTEST( N, LAST, BR, I1, I2 )                                  100
      INTEGER BR                                                                200
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL          300
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(13)                              400
   10 KOL = LAST                                                                500
      IF ( KARD(LAST+1) .EQ. KSPK(8) ) GO TO 50                                 600
      IF ( KARD(LAST+1) .NE. KBL ) GO TO 200                                    700
      LAST = LAST + 1                                                           800
      GO TO 10                                                                  900
   50 LAST = LAST+2                                                            1000
      KARD(LAST-1) = KBL                                                       1100
   60 IF ( KARD(LAST) .EQ. KDIG(I1+1) ) GO TO 150                              1200
      IF ( KARD(LAST) .EQ. KDIG(I2+1) ) GO TO 150                              1300
      IF ( I2 .LT. 10 ) GO TO 99                                               1400
      I1 = I2/10                                                               1500
      I2 = I2 - 10*I1                                                          1600
      GO TO 60                                                                 1700
   99 CALL ERROR                                                               1800
      RETURN                                                                   1900
  100 LAST = LAST + 1                                                          2000
      GO TO 60                                                                 2100
  150 KARD(LAST) = KBL                                                         2200
      KMAX = KMAX - 2                                                          2300
      DO 170 I=KOL,KMAX                                                        2400
  170   KARD(I+1) = KARD(I+3)                                                  2500
  200 CONTINUE                                                                 2600
      IF ( BR .EQ. 1 ) RETURN 1                                                2700
      RETURN                                                                   2800
        END                                                                    2900
```

```
      SUBROUTINE UPDATE                                                 100
C                                                                       200
C     THIS SUBROUTINE UPDATES THE SYMBOL TABLE.  NEW SYMBOLS ARE ADDED  300
C     AS THEY ARE FOUND.  REFERENCES TO PREVIOUSLY FOUND SYMBOLS ARE    400
C     UPDATED.                                                          500
C                                                                       600
      LOGICAL*1 O1, OTYPE(4), ONAME(8), FORCEM, FORCES                  700
      LOGICAL OUPDAT, KSBO, KCDO                                        800
      REAL*8 NAME(2), SYMTAB(2,800)                                     900
      INTEGER*2 T1,T2,TT(2),N2,NN(4),N3                                1000
      INTEGER*4 JLOC(400), CTAB(400),TEMP,HTAB(1)                      1100
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL,K 1200
     1RSX,LIST(8191),LOCN,MINA,MAXA,NCHAR,NEXT,NRT,NREC,NROUT,NSYM     1300
     2  ,NERR,NERO,MEMKEY                                              1400
      COMMON /ALPH/ KBL,KABC(26),KDIG(10),KSPK(11)                     1500
      COMMON /BETA/ MXCH,MXLI,N15B,LINE,NPAGE,KSBO,KCDO                1600
      COMMON / HASH / ILOC, LOC, NAME, IVAL                            1700
      COMMON /XUPDAT/ OUPDAT, ICDICT(27)                               1800
C                                                                      1900
      EQUIVALENCE (TEMP,TT(1)),(TT(1),T1),(TT(2),T2)                   2000
      EQUIVALENCE ( NAME(1),O1,N1,NN(1) ), ( NN(3),N2 ), ( OTYPE(1),KEY) 2100
      EQUIVALENCE ( SYMTAB(1,1), LIST(4192) )                          2200
     1    , ( HTAB(1),LIST(1) ), ( NN(4),N3 ), ( JLOC(1),LIST(7392) )  2300
      DATA NST /  800 /                                                2400
C                                        HASH THE NAME                 2500
      IF ( KODE .LT. MINA  .OR.  KODE .GT. MAXA ) RETURN               2600
      FORCES = .FALSE.                                                 2700
      FORCEM = .FALSE.                                                 2800
      IADD = MOD( IABS( N1+N2+N3 ), 4191 )                             2900
      TEMP = HTAB(IADD+1)                                              3000
      IFLAG = IABS( ITYPE )                                            3100
C-----------------------  ITYPE                                       3200
C               3     CHARACTER                                        3300
C               5     COMMON              12     INTEGER               3400
C               6     COMPLEX             13     LOGICAL               3500
C               8     DIMENSION           18     REAL                  3600
C                                                                      3700
      IF ( IFLAG .GT. 18 ) GO TO 25                                    3800
      GO TO ( 25,25, 3,25, 5, 6,25, 8,25,25,25,12,13,25,25,25,25,18),  3900
     *        IFLAG                                                    4000
    3 KEY = IVAL + 5                                                   4100
      GO TO 30                                                         4200
    5 KEY = 1                                                          4300
      GO TO 40                                                         4400
    6 KEY = 2                                                          4500
      GO TO 30                                                         4600
    8 KEY = 1                                                          4700
      GO TO 40                                                         4800
   12 KEY = 3                                                          4900
      GO TO 30                                                         5000
   13 KEY = 4                                                          5100
      GO TO 30                                                         5200
   18 KEY = 5                                                          5300
      GO TO 30                                                         5400
   25 KEY = 0                                                          5500
```

```
      GO TO 50                                                                  5600
   30 FORCEM = .TRUE.                                                           5700
   40 IF ( KRSX .EQ. KSPK(3) ) FORCES = .TRUE.                                  5800
   50 CONTINUE                                                                  5900
C                                        IS THIS A NEW NAME                    6000
      IF(T1.NE.0)GO TO 100                                                      6100
      ILOC = ILOC+1                                                             6200
      IF(ILOC.GT.NST)GO TO 500                                                  6300
      LOC = ILOC                                                                6400
      T1  = LOC                                                                 6500
      HTAB(IADD+1) = TEMP                                                       6600
      GO TO 270                                                                 6700
C                                        HAVE WE FOUND THE NAME                6800
  100 IF ( NAME(1) .NE. SYMTAB(1,T1) ) GO TO 200                                6900
      IF ( NAME(2) .NE. SYMTAB(2,T1) ) GO TO 200                                7000
  120 IF ( FORCEM ) JLOC(T1) = ISIGN( KEY, JLOC(T1) )                           7100
      IF ( FORCES ) JLOC(T1) = -MAX0( 1, IABS( JLOC(T1) ) )                     7200
      LOC = JLOC(T1)                                                            7300
  150 RETURN                                                                    7400
C                                        DO WE START A CHAIN                   7500
  200 IF(T2.NE.0)GO TO 300                                                      7600
      ICOL = ICOL+1                                                             7700
      IF ( ICOL .GT. 400 ) GO TO 500                                            7800
      T2   = ICOL                                                               7900
      HTAB(IADD+1) = TEMP                                                       8000
      ILOC = ILOC+1                                                             8100
      IF(ILOC.GT.NST)GO TO 500                                                  8200
  250 LOC = ILOC                                                                8300
      T1  = LOC                                                                 8400
      T2  = 0                                                                   8500
      CTAB(ICOL)  = TEMP                                                        8600
  270 SYMTAB(1,LOC) = NAME(1)                                                   8700
      SYMTAB(2,LOC) = NAME(2)                                                   8800
      IF ( FORCES ) KEY = -KEY                                                  8900
      JLOC(T1) = KEY                                                            9000
  290 LOC = KEY                                                                 9100
      RETURN                                                                    9200
C                                        SEARCH IN CHAIN                       9300
  300 ITEM = T2                                                                 9400
      TEMP = CTAB(ITEM)                                                         9500
      IF ( NAME(1) .NE. SYMTAB(1,T1) ) GO TO 400                                9600
      IF ( NAME(2) .EQ. SYMTAB(2,T1) ) GO TO 120                                9700
C                                        HAVE WE FOUND THE NAME                9800
C                                        DO WE ADD TO THE CHAIN                9900
  400 IF(T2.NE.0)GO TO 300                                                     10000
      ICOL = ICOL+1                                                            10100
      IF ( ICOL .GT. 400 ) GO TO 500                                           10200
      T2   = ICOL                                                              10300
      CTAB(ITEM) = TEMP                                                        10400
      ILOC = ILOC+1                                                            10500
      IF(ILOC.LE.NST)GO TO 250                                                 10600
C                                        OVERFLOW                             10700
  500 WRITE(6,501)                                                             10800
  501 FORMAT(16HOHASH TABLE FULL)                                              10900
      STOP                                                                     11000
```

```
C                          INITIALIZE OR DISPLAY                          11100
      ENTRY HASHIN                                                        11200
      CALL SORT( 0 )                                                      11300
      OUPDAT = .FALSE.                                                    11400
      KSBO = .TRUE.                                                       11500
      JSYM = 0                                                            11600
      DO 600 I=1,4191                                                     11700
  600   HTAB(I) = 0                                                       11800
      DO 620 I=1,400                                                      11900
  620   CTAB(I) = 0                                                       12000
      ILOC = 0                                                            12100
      ICOL = 0                                                            12200
      RETURN                                                              12300
        END                                                               12400
```

```
      SUBROUTINE XTRNAL                                                 100
      COMMON ITYPE,JOB(66),KARD(1326),KBUFF(80),KLEAR(8),KMAX,KODE,KOL  200
C------ TEST SUBPROGRAMS IN AN EXTERNAL STATEMENT FOR VALIDITY IN       300
C           6600 FORTRAN.                                               400
   11 KSAVE = KOL                                                       500
      CALL SYMBOL                                                       600
      IF ( KODE .LE. 0 ) RETURN                                         700
      K = KOL                                                           800
      KOL = KSAVE                                                       900
      CALL FNCHEK( -1 )                                                1000
      KOL = K                                                          1100
      GO TO 11                                                         1200
        END                                                            1300
```

```
         MACRO
&LABEL   SET
         TITLE '                 SUBROUTINE   &LABEL'
&LABEL   SAVE  (14,12),,*
         LA    0,&LABEL-IO .     LOCAL DISPLACEMENT ---> GR#0
         SR    15,0 .            POINT GR 15 BACK TO THE CSECT
         LR    10,13 .           USERS SAVE AREA ADDRESS ---> TEMP STORAGE
         LR    13,15 .           GLOBAL ADDRESSABILITY IN GR 13
         ST    13,8(10) .        ESTABLISH TRACE BACK CHAIN
         ST    10,4(13) .        ESTABLISH VALID BACK CHAIN
         MEND
         SPACE 3
IO       START
         DS    18F
         ENTRY READ,WRITE,TREAD,TWRITE,OPEN,CLOSE,TRANCE
         ENTRY ICHAR
         ENTRY TRNSL8,MOVTMP,SETTMP
         SPACE 3
         ENTRY L4CV6029
         USING *,15
L4CV6029 LR    0,1
         LA    1,255
LOOP29   STC   1,TAB(1)
         BCT   1,LOOP29
         LR    1,0
         L     15,=V(MAIN)
         BR    15
         SPACE 3
         USING *,15
SETTMP   LA    1,TEMP
         ST    1,DCBAD
         BR    14
         DROP  15
         USING IO,13
OPEN     SET
         OPEN  (INPUT,,OUTPUT,(OUTPUT),TEMP,(OUTPUT))
         LA    15,0
         TM    INPUT+48,X'10'
         BZ    OPNERR
         TM    OUTPUT+48,X'10'
         BZ    OPNERR
         TM    TEMP+48,X'10'
         BZ    OPNERR
         LA    1,OUTPUT
         ST    1,DCBAD
ROPN     L     13,4(13)
         RETURN (14,12),T,RC=(15)
OPNERR   LA    15,4
         B     ROPN
CLOSE    SET
         CLOSE (INPUT,,OUTPUT,,TEMP,)
         LA    15,0
         B     ROPN
READ     SET
         L     11,0(1)           FETCH ARRAY ADDRESS
```

```
         GET    INPUT,(11)
         TR     0(80,11),INTAB
         LA     1,80
         LA     2,316(11)
         LA     3,79(11)
         L      5,=CL4' '
         L      6,=F'4'
A1TOA4   IC     4,0(3)
         SRDL   4,8
         ST     5,0(2)
         BCTR   3,0
         SR     2,6
         SLL    5,8
         BCT    1,A1TOA4
         LA     15,0
         B      ROPN
WRITE    SET
         L      11,0(1)
         PUT    OUTPUT,(11)
         LA     15,0
         B      ROPN
TREAD    SET
         L      11,0(1)
         GET    TEMP,(11)
         LA     15,0
         B      ROPN
TWRITE   SET
         L      11,0(1)
         PUT    TEMP,(11)
         LA     15,0
         B      ROPN
TRANCE   SET
         LM     2,3,0(1)           GET ARGUMENTS
         LA     12,BUF
         L      9,0(3)             GET NUMBER OF CHARACTERS TO BE OUTPUT
         LTR    9,9                NUMBER MUST BE POSITIVE.
         BP     NINEOK             IF NOT,
         LA     9,72                        SET IT TO OUTPUT 1 CARD IMAGE
NINEOK   LA     11,0(12)              ADDRESS OF AREA BEING PACKED
         LA     7,72               NUMBER OF COLUMNS TO BE PACKED (MAX)
         XR     3,3
LOOP     AR     3,9
         STC    3,MBL+3            IF NOT FULL CARD, LOCATION TO BEGIN BLAN
         LR     1,9                NUMBER OF CHARACTERS REMAINING TO BE PAK
         SR     9,7                TEST FOR FULL CARD
         BM     A4TOA1             BRANCH IF NOT FULL CARD
         LR     1,7                ELSE SET NUMBER OF COLUMNS TO PACK
A4TOA1   IC     3,0(2)
         STC    3,0(11)
         LA     11,1(11)
         LA     2,4(2)
         BCT    1,A4TOA1
         MVC    72(8,12),BLANKS
         LTR    9,9                TEST TO SEE IF BLANKS NEEDED AT END OF ST
         BNM    TRPUT
```

```
       |  |LPR    1,9            NUMBER OF BLANKS NEEDED
       |  |EX     1,MBL          BLANK OUT RIGHTMOST COLS OF CARD
TRPUT  |  |L      1,DCBAD
       |  |PUT    (1),(12)
       |  |LTR    9,9            TEST FOR TASK COMPLETE
       |  |BNP    ROPN           ALL DONE UNLESS R9 > 0
TRMVC  |  |MVC    0(6,12),CONT
       |  |LA     11,6(12)       SET UP FOR CONTINUATION CARDS
       |  |LA     7,66           66 COLUMNS IN A CONTINUATION CARD
       |  |LA     3,6
       |  |B      LOOP
MBL    |  |MVC    0(0,12),BLANKS
BLANK  |S |   DC     15C'    '      DON'T CHANGE ORDER OF THESE TWO CARDS  1
CONT   |  |   DC     C'      +'     DON'T CHANGE ORDER OF THESE TWO CARDS  2
ICHAR  |  |SET
       |  |LM     2,3,0(1)
       |  |L      3,0(3)
       |  |SR     4,4
NXT    |  |IC     4,0(2)
       |  |IC     4,CHAR(4)
       |  |STH    4,2(2)
       |  |LA     2,4(2)
       |  |BCT    3,NXT
       |  |LA     15,0
       |  |B      ROPN
MOVTM  |P |   SET
       |  |L      11,0(1)
       |  |CLOSE (TEMP,DISP)
IN     |  |OPEN (TEMP,(INPUT))
       |  |TM   TEMP+48,X'10'
       |  |BZ   OPNERR
COPYL  |O |OP GET    TEMP,(11)
       |  |PUT    OUTPUT,(11)
       |  |B      COPYLOOP
COPYD  |O |NE LA     1,OUTPUT
       |  |ST     1,DCBAD
       |  |CLOSE (TEMP,DISP)
       |  |OPEN (TEMP,(OUTPUT))
       |  |TM   TEMP+48,X'10'
       |  |BZ   OPNERR
       |  |LA     15,0
       |  |B      ROPN
TRNSL  |S |   SET
       |  |L      11,0(1)
       |  |TR     0(80,11),TAB
       |  |PUT    OUTPUT,(11)
       |  |LA     15,0
       |  |B      ROPN
       |  |EJECT
TAB    |  |DC       256X'5D'          UP ARROW
ENDTA  |B |   EQU    *
       |  |ORG      TAB+C'A'
       |  |DC       C'ABCDEFGHI'
       |  |ORG      TAB+C'J'
       |  |DC       C'JKLMNOPQR'
```

```
         ORG   TAB+C'S'                                                  16600
         DC    C'STUVWXYZ'                                              16700
         ORG   TAB+C'0'                                                 16800
         DC    C'0123456789'                                            16900
         ORG   TAB+C'.'                                                 17000
*        . < ( + | &          360 CHAR                                  17100
         DC    X'4BC06C505D50'          6600                            17200
         ORG   TAB+C' '                                                 17300
         DC    C' '                                                     17400
         ORG   TAB+X'5A'                                                17500
*        ! $ * ) ; @ - /      360 CHAR                                  17600
         DC    X'5E5B5C4C6B4E6061'      6600                            17700
         ORG   TAB+X'6B'                                                17800
*        , % _ > ?            360 CHAR                                  17900
         DC    X'6B5D5D5F5D'            6600                            18000
         ORG   TAB+X'7A'                                                18100
*        : # @ ' = "          360 CHAR                                  18200
         DC    X'7AD06D7C7B6F'          6600                            18300
         ORG   ENDTAB                                                   18400
         SPACE 3                                                        18500
CHAR     DC    256AL1(49)                                               18600
CHARE ND EQU   *                                                        18700
         ORG   CHAR+C'A'                                                18800
*        A B C D E F G H I                                              18900
         DC    X'010203040506070809'                                    19000
         ORG   CHAR+C'J'                                                19100
*        J K L M N O P Q R                                              19200
         DC    X'0A0B0C0D0E0F101112'                                    19300
         ORG   CHAR+C'S'                                                19400
*        S T U V W X Y Z                                                19500
         DC    X'131415161718191A'                                      19600
         ORG   CHAR+C'0'                                                19700
*        0 1 2 3 4 5 6 7 8 9                                            19800
         DC    X'1B1C1D1E1F2021222324'                                  19900
         ORG   CHAR+C'='                                                20000
         DC    AL1(37)                                                  20100
         ORG   CHAR+C','                                                20200
         DC    AL1(38)                                                  20300
         ORG   CHAR+C'('                                                20400
         DC    AL1(39)                                                  20500
         ORG   CHAR+C'/'                                                20600
         DC    AL1(40)                                                  20700
         ORG   CHAR+C')'                                                20800
         DC    AL1(41)                                                  20900
         ORG   CHAR+C'+'                                                21000
         DC    AL1(42)                                                  21100
         DC    AL1(49)                                                  21200
         DC    AL1(42)                                                  21300
         ORG   CHAR+C'-'                                                21400
         DC    AL1(43)                                                  21500
         ORG   CHAR+C'*'                                                21600
         DC    AL1(44)                                                  21700
         ORG   CHAR+C'.'                                                21800
         DC    AL1(45)                                                  21900
         ORG   CHAR+C'$'                                                22000
```

```
         DC      AL1(46)                                                        22100
         ORG     CHAR+X'7D'                                                     22200
         DC      AL1(47)                                                        22300
         ORG     CHAR+C' '                                                      22400
         DC      AL1(48)                                                        22500
         ORG     CHAREND                                                        22600
         SPACE 3                                                                22700
INTAB    DC        256X'5D'                                                     22800
INTEND   EQU     *                                                             22900
         ORG   INTAB+C'A'                                                       23000
         DC      C'ABCDEFGHI'                                                   23100
         ORG   INTAB+C'J'                                                       23200
         DC      C'JKLMNOPQR'                                                   23300
         ORG   INTAB+C'S'                                                       23400
         DC      C'STUVWXYZ'                                                    23500
         ORG   INTAB+C'0'                                                       23600
         DC      C'0123456789'                                                  23700
         ORG   INTAB+C'.'                                                       23800
*        . ) ( + | &                                                            23900
         DC      X'4B5D4D4E4F50'                                                24000
         ORG   INTAB+X'5A'                                                      24100
*        ! $ * ) ; @ - /                                                        24200
         DC      X'5A5B5C5D5E5F6061'                                            24300
         ORG   INTAB+X'6B'                                                      24400
*        , ( _ > ?                                                              24500
         DC      X'6B4D6D6E6F'                                                  24600
         ORG   INTAB+X'7A'                                                      24700
*        : = ' ' = "                                                            24800
         DC      X'7A7E7D7D7E7F'                                                24900
         ORG   INTAB+X'40'                                                      25000
         DC      C' '                                                           25100
         ORG     INTEND                                                         25200
         SPACE 3                                                                25300
DCBAD    DS      1F                                                             25400
BUF      DS      20F                                                            25500
         EJECT                                                                  25600
INPUT    DCB     DDNAME=INPUT,DSORG=PS,MACRF=(GM),LRECL=80,RECFM=FB,      X     25700
         DEVD=DA,EODAD=OPNERR                                                   25800
OUTPUT   DCB     DDNAME=OUTPUT,DSORG=PS,MACRF=(PM),LRECL=80,RECFM=FB,     X     25900
         DEVD=DA                                                                26000
TEMP     DCB     DDNAME=TEMP,DSORG=PS,MACRF=(GM,PM),LRECL=80,RECFM=FB,    X     26100
         DEVD=DA,EODAD=COPYDONE                                                 26200
           END                                                                  26300
```

ROUTINE NAME	ROUTINE TYPE	LOCATION
MAIN	PROGRAM	SCOFF.TMP
BLOCK DATA	BLOCK DATA	SCOFF.TMP
CENTOR	SUBROUTINE	SCOFF.TMP
ERREND	SUBROUTINE	SCOFF.TMP
ERROR	SUBROUTINE	SCOFF.TMP
ERR	SUBROUTINE	SCOFF.TMP
FNCHEK	SUBROUTINE	SCOFF.TMP
FORTRN	SUBROUTINE	SCOFF.TMP
IMPSET	SUBROUTINE	SCOFF.TMP
INPUT	SUBROUTINE	SCOFF.TMP
INSRTR	SUBROUTINE	SCOFF.TMP
MOVER	SUBROUTINE	SCOFF.TMP
MULTEQ	SUBROUTINE	SCOFF.TMP
PAG	SUBROUTINE	SCOFF.TMP
RETRNI	SUBROUTINE	SCOFF.TMP
SIXFR7	SUBROUTINE	SCOFF.TMP
SUB6	SUBROUTINE	SCOFF.TMP
SLASHS	SUBROUTINE	SCOFF.TMP
SORT	SUBROUTINE	SCOFF.TMP
STARL	SUBROUTINE	SCOFF.TMP
SUBSUB	SUBROUTINE	SCOFF.TMP
SYMBOL	SUBROUTINE	SCOFF.TMP
TDTEST	SUBROUTINE	SCOFF.TMP
UPDATE	SUBROUTINE	SCOFF.TMP
	ENTRY POINT	UPDATE
XTRNAL	SUBROUTINE	SCOFF.TMP
MAIN	PROGRAM	SCOFF.TMP
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN

ROUTINE NAME	ROUTINE TYPE	LOCATION
	ENTRY POINT	UPDATE
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN
	ENTRY POINT	MAIN
CENTOR	SUBROUTINE	SCOFF.TMP
ERR	SUBROUTINE	SCOFF.TMP
ERREND	SUBROUTINE	SCOFF.TMP
ERROR	SUBROUTINE	SCOFF.TMP
FNCHEK	SUBROUTINE	SCOFF.TMP
FORTRN	SUBROUTINE	SCOFF.TMP
IMPSET	SUBROUTINE	SCOFF.TMP
INPUT	SUBROUTINE	SCOFF.TMP
INSRTR	SUBROUTINE	SCOFF.TMP
MAIN	PROGRAM	SCOFF.TMP
MAIN	PROGRAM	SCOFF.TMP
MOVER	SUBROUTINE	SCOFF.TMP
MULTEQ	SUBROUTINE	SCOFF.TMP
PAG	SUBROUTINE	SCOFF.TMP
RETRNI	SUBROUTINE	SCOFF.TMP
SIXFR7	SUBROUTINE	SCOFF.TMP
SLASHS	SUBROUTINE	SCOFF.TMP
SORT	SUBROUTINE	SCOFF.TMP
STARL	SUBROUTINE	SCOFF.TMP
SUB6	SUBROUTINE	SCOFF.TMP
SUBSUB	SUBROUTINE	SCOFF.TMP
SYMBOL	SUBROUTINE	SCOFF.TMP
TDTEST	SUBROUTINE	SCOFF.TMP
UPDATE	SUBROUTINE	SCOFF.TMP
XTRNAL	SUBROUTINE	SCOFF.TMP
BLOCK DATA	BLOCK DATA	SCOFF.TMP

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Armed Forces Radiobiology Rsch Institute
2 cy ATTN: Dir

Armed Forces Staff College
ATTN: Library

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Executive Assistant

Defense Communications Engineer Ctr
ATTN: R-123

Defense Intelligence Agency
2 cy ATTN: RTS-2A

Defense Nuclear Agency
ATTN: RAAE
ATTN: SPAS
ATTN: RAEV
ATTN: NATA
ATTN: RAEE
ATTN: SPSS
ATTN: SPTD
ATTN: NATD
ATTN: NATA, J. Anderson
ATTN: COMP-1, B. Robertson
ATTN: STTI, M. Vincent
2 cy ATTN: STTI
4 cy ATTN: TITL

Defense Tech Info Ctr
12 cy ATTN: DD

Field Command Defense Nuclear Agency
Det 1
Lawrence Livermore Lab
ATTN: FC-1, J. Crandley
ATTN: FC-1

Field Command
Defense Nuclear Agency
ATTN: FCTXE
ATTN: FCPR
ATTN: FCTT, S. Humphries
ATTN: FCTT, W. Summa
ATTN: FCTT, G. Ganong
ATTN: FCTT

Joint Chiefs of Staff
ATTN: SAGA/SFD
ATTN: SAGA/STA
ATTN: J-5, Nuclear Div/Strategy Div
ATTN: J-3

Joint Strat Tgt Planning Staff
ATTN: JLA, Threat Applications Div
2 cy ATTN: NRI-STINFO, Library

Joint Strategic Connectivity Staff
ATTN: JCS/DNA, F. Tedesco

Under Secretary of Defense for Rsch & Engrg
ATTN: Strategic & Space Sys (OS)

DEPARTMENT OF THE ARMY

Aberdeen Proving Ground
2 cy ATTN: Tech Library

Atmospheric Sciences Lab
ATTN: DELAS-AS

BMD Advanced Technology Ctr
ATTN: ATC-T

Deputy Chief of staff for Ops & Plans
ATTN: DAMO-NCN

Deputy Chief Staff for Rsch Dev & Acq
ATTN: DAMA-CSS-N

Engr Studies Ctr
ATTN: DAEN-FES

Harry Diamond Labs
ATTN: DELHD-NW-P, 20240
ATTN: DELHD-TA-L, 81100
ATTN: DELHD-NP

Multi Service Communications Systems
ATTN: DRCPM-MSCS-APB, M. Francis

US Army Ballistic Rsch Labs
ATTN: R. Reisler
2 cy ATTN: DRDAR-TSB-S

US Army Chemical School
ATTN: ATZN-CM-CC

US Army Comb Arms Combat Dev Acty
ATTN: ATZL-CAN-I

US Army Comd & General Staff College
ATTN: ACQ, Library Div

US Army Communications Command
ATTN: Tech Ref Div

US Army Concepts Analysis Agency
ATTN: Classified Documents

US Army Engr Waterways Exper Station
ATTN: J. Houston
ATTN: WESGH, P. Hadala

US Army Field Artillery School
ATTN: M. Swett, Library

US Army Forces Command
ATTN: Tech Library

US Army Intell & Sec Cmd
ATTN: Tech Library

US Army Logistics Ctr
ATTN: ATCL-OSS, S. Cockrell

US Army Material Dev & Readiness Cmd
ATTN: DRCDE-D

DEPARTMENT OF THE ARMY (Continued)

US Army Materiel Sys Analysis Actvy
ATTN: DRXSY-CC

US Army Nuclear & Chemical Agency
ATTN: Library

US Army Satellite Comm Agency
ATTN: Tech Library

US Army Signal Ctr & Ft Gordon
ATTN: ATZH-CD

US Army TRADOC Sys Analysis Actvy
ATTN: ATAA-PL

US Army War College
2 cy ATTN: Library

US Army White Sands Missile Range
ATTN: Commander

USA Missile Command
ATTN: Doc Section

DEPARTMENT OF THE NAVY

Chief of Naval Operations
Dir Cmd Control Planning & Prgmng Div, OP-940
ATTN: OP-943

David Taylor Naval Ship R&D Ctr
5 cy ATTN: Code L42-3

Naval Air Development Ctr
ATTN: Doc Con

Naval Air Systems Cmd
ATTN: Tech Library

Naval Civil Engrg Lab
ATTN: L-51, S. Johnson

Naval Electronic Systems Cmd
ATTN: Tech Library

Naval Ocean Systems Ctr
ATTN: Code 4471

Naval Postgraduate School
2 cy ATTN: Code 1424, Library

Naval Rsch Lab
5 cy ATTN: Code 2627

Naval Space Surveillance System
ATTN: J. Burton

Naval Surface Weapons Ctr
ATTN: Code R44, H. Glaz
ATTN: Code F31

Naval Surface Weapons Ctr
5 cy ATTN: Tech Library & Info Svcs Br

Naval Weapons Ctr
5 cy ATTN: Code 343, FKA6A2, Tech Svcs

DEPARTMENT OF THE NAVY (Continued)

Naval Weapons Evaluation Facility
ATTN: H. Struve

Naval Underwater Systems Ctr
2 cy ATTN: Code EM, J. Kalinowski

Office of Naval Rsch
ATTN: Tech Info Svcs

DEPARTMENT OF THE AIR FORCE

Air Force Propulsion Lab
ATTN: LKDH, Stop 24, E. Haberman

Air Force Radiobiological Health Lab, AFLC
ATTN: Dir

Air Force Armament Lab
ATTN: AFATL/DLY

Air Force Rocket Propulsion Lab
ATTN: LKCP

Air Force Tech Applications Ctr
2 cy ATTN: STINFO Ofc-TF

Air Force Test & Evaluation Ctr
ATTN: J. Hoge

Air Force Weapons Lab
ATTN: AD, J. Ormand
ATTN: SUL
ATTN: NTES, R. Guice
ATTN: NTYC, J. Burgio

Air Force Wright Aeronautical Lab
ATTN: Tech Library

Air Force Wright Aeronautical Lab
ATTN: LTE

Air University Library
ATTN: AUL-LSE

Headquarters
Air Weather Service, MAC
ATTN: AWSAE

Armament Dev & Test Ctr
ATTN: Doc Con

Ballistic Missile Office
ATTN: SYDT
ATTN: EN
ATTN: ENMP, D. Van Gari

Deputy Chief of Staff
Rsch, Dev, & Acq
ATTN: AFRDST
ATTN: AFRDS, Space Sys & C3 Dir
ATTN: AFRDQI

Headquarters
Electronic Systems Div
ATTN: DCK, L. Staples
ATTN: XRRT

DEPARTMENT OF THE AIR FORCE (Continued)

Foreign Tech Div
ATTN: NIIS, Library

Headquarters
NORAD
ATTN: J5YS, F. Smith

Space Div
ATTN: IND
ATTN: SSP-6
ATTN: YGJ
ATTN: YKM

DEPARTMENT OF ENERGY

Department of Energy
Tech Info Ctr
ATTN: Doc Con

OTHER GOVERNMENT AGENCIES

Central Intell Agency
ATTN: OSWR/NED

Department of the Interior
US Geological Survey
ATTN: J. Dietrich

Department of the Interior, US Geological Survey
ATTN: R. Regan

Federal Emergency Management Agency
ATTN: Ofc of Rsch/NP, D. Bensen

DEPARTMENT OF ENERGY CONTRACTORS

University of California
Lawrence Livermore National Lab
ATTN: L-21, D. Oakley
ATTN: P. Colella
ATTN: Tech Info Dept, Library

Los Alamos National Labratory
ATTN: Reports Library

Sandia National Labs, Livermore
ATTN: Library & Security Classification Div

DEPARTMENT OF DEFENSE CONTRACTORS

Aerospace Corp
ATTN: M. Eskijian

Agbabian Associates
ATTN: M. Agbabian

Analytic Services, Inc
2 cy ATTN: Library

Applied Rsch Associates, Inc.
ATTN: J. Phillips

Applied Theory, Inc.
ATTN: Doc Con

APTEK, Inc.
ATTN: A. Larrabee

Astron Rsch & Engrg
ATTN: A. Levien

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

AVCO Systems Div
ATTN: Library, A830

BDM Corp
ATTN: J. Braddock
ATTN: M. Kelly
ATTN: L. Schlipper
2 cy ATTN: Corporate Library

BDM Corp.
ATTN: P. McClousky
ATTN: Library
ATTN: D. Peercy
ATTN: F. Leech

Beers Associates, Inc.
ATTN: Contracts Ofc

Berkeley Rsch Associates, Inc
ATTN: J. Workman

Boeing Co.
ATTN: Aerospace Library
ATTN: R. Hager
ATTN: W. Cooley
ATTN: S. Sandberg
ATTN: E. Brown
4 cy ATTN: M/S 42-37, B. Henderson

Boeing Computer Svcs, Inc.
ATTN: M/S 8C-52, D. Chapman
ATTN: B. Brown

Boeing Military Airplane Co.
ATTN: D. Pierson
ATTN: R. Syring

Booz-Allen & Hamilton, Inc
ATTN: R. Hanson

California Rsch & Tech, Inc
ATTN: T. Isabelle
ATTN: K. Kreyenhagen
ATTN: Library

Calspan Corp.
ATTN: A. Davis

Calspan Corp.
ATTN: M. Dunn
ATTN: Library

66th MI Group
ATTN: RDA

Control Data Corp.
ATTN: R. Portillo

Corrales Applied Physics Co.
ATTN: N. Froula

Cushing Associates
ATTN: V. Cushing

Decision Science Consortium, Inc.
ATTN: J. Chinnis

Dikewood
ATTN: Tech Library for K. Sabisch

Dynalectron
ATTN: C. Scott

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

EG&G Wash Analytical Svcs Ctr, Inc
ATTN: Library
ATTN: J. Bridges
ATTN: J. Ridge
ATTN: L. Daley

Electro-Magnetic Applications, Inc
ATTN: B. Fisher

Electro-Mech Systems, Inc
ATTN. A. Stough
ATTN: R. Shunk

Electrospace Systems, Inc
ATTN: H. Logston

ESL, Inc
ATTN: Library

Flow Sciences, Inc
ATTN: J. Campbell

Garjak Rsch, Inc
ATTN: G. Erickson

General Electric Co
ATTN: N. Dispensiere
ATTN: D. Coceano
ATTN: Tech Info Ctr

General Research Corp
ATTN: J. Garbarino
ATTN: G. Purvis
ATTN: D. Sangster
ATTN: J. Balter
ATTN: L. Wogulis
2 cy ATTN: Doc Con
2 cy ATTN: Tech Info Ofc

Harold Rosenbaum Associates, Inc
ATTN: H. Rosenbaum

Horizons Technology, Inc
ATTN: R. Kruger

Information Science, Inc
ATTN: W. Dudziak

Institute for Defense Analyses
ATTN: Classified Library

IRT Corp
ATTN: M. Obrien
2 cy ATTN: Library

J. D. Haltiwanger Consulting Svcs
ATTN: Doc Con

JAYCOR
ATTN: W. Hobbs
ATTN: W. Radaski

JAYCOR
ATTN: E. Wenaas
ATTN: A. Woods
ATTN: J. Young
ATTN: A. J. Woods
5 cy ATTN: Library

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Johns Hopkins University
ATTN: F. Brelsford

Kaman Avidyne
ATTN: M. Tomayko
2 cy ATTN: Library

Kaman Sciences Corp
ATTN: D. Conroy
ATTN: F. Shelton
5 cy ATTN: Library

Kaman Tempo
ATTN: K. Schwartz
ATTN: DASIAC
ATTN: C. Anderson
ATTN: A. Klein
10 cy ATTN: W. Chan

Kaman Tempo
ATTN: R. Miller
ATTN: Contracts Ofc

K Tech Corp
ATTN: B. Mason
ATTN: E. Young

JAYCOR
2 cy ATTN: Library

JAYCOR
ATTN: C. Clark

Lockheed Missiles & Space Co, Inc
ATTN: J. Henley
ATTN: T. Geers
2 cy ATTN: Reports Library

Lockheed Missiles & Space Co, Inc
ATTN: C. Rankin
2 cy ATTN: TIC-Library

Martin Marietta Denver Aerospace
ATTN: Rsch Library, 6617

Mathematical Sciences Northwest, Inc
ATTN: Doc Con

Martin Marietta Corp
2 cy ATTN: M. Griffith

Maxwell Labs, Inc
ATTN: J. Smith
ATTN: A. Kolb

McDonnell Douglas Corp
ATTN: M. Gee
ATTN: W. Gee
ATTN: D. Dean
2 cy ATTN: Tech Library Svcs

McDonnell Douglas Corp
ATTN: R. Mapes

Merritt CASES, Inc
ATTN: Library
ATTN: G. Wintergerst

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

University of Miami
ATTN: Contract Ofc, S. Wang

Mission Research Corp
ATTN: A. Gregerson
ATTN: C. Longmire
5 cy ATTN: Doc Con

Mission Research Corp
ATTN: Library
ATTN: B. Neri

Mitre Corp
ATTN: Tech Report Ctr

Mitre Corp
ATTN: A. Schneider

Northrop Corp
ATTN: A. Scott
2 cy ATTN: RAD Effects Grp

Pacific-Sierra Rsch Corp
ATTN: S. Sanemitser
ATTN: J. Levsque
ATTN: S. Finn
ATTN: Contracts Ofc
ATTN: H. Brode, Chairman SAGE
ATTN: Doc Con

Pacific-Sierra Research Corp
ATTN: D. Gormley
ATTN: Doc Con

Pacifica Technology
ATTN: G. Kent
ATTN: M. Gittings
2 cy ATTN: Tech Library

PDA Engrg
2 cy ATTN: Doc Con

Physical Dynamics, Inc
ATTN: E. Fremouw

Physical Dynamics, Inc
ATTN: Contracts Ofc

Physics International Co
ATTN: J. Larson
ATTN: J. Shea
ATTN: L. Burr
5 cy ATTN. Tech Library

Pulse Sciences Inc
ATTN: W. Weseloh

R & D Associates
ATTN: F. Field
ATTN: G. Ivy
ATTN: M. Wright
ATTN: M. Warshaw
10 cy ATTN: Tech Info Ctr

R & D Associates
2 Cy ATTN: Doc Con

R. F. Cross Associates, Ltd
ATTN: R. Mahoney

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Rand Corp
2 cy ATTN: Library

Rockwell International Corp
ATTN: Library

S-CUBED
ATTN: J. Glow
ATTN: M. Scarpa
ATTN: K. Pyatt
10 cy ATTN: Doc Con
10 Cy ATTN: Library

S-CUBED
ATTN: J. Murphy

S-CUBED
ATTN: J. Crepeau

Santa Fe Corp
2 cy ATTN: Sec Ofc
2 cy ATTN: D. Paolucci

Science & Engrg Associates, Inc
ATTN: J. Prentice

Science & Engrg Associates, Inc
ATTN: J. McCullough

Science Applications, Inc
ATTN: R. Reaugh

Science Applications, Inc
ATTN: C. Dougherty

Science Applications, Inc
ATTN: B. Dial

Science Applications, Inc
ATTN: C. Burgart
10 cy ATTN: Sec Ofc

Science Applications, Inc
ATTN: Sec Ofc
ATTN: Contracts Ofc

Science Applications, Inc
ATTN: Contracts Ofc
ATTN: B. Elyea, Security Ofc

Science Applications, Inc
ATTN: Security Ofc
ATTN: B. Goplen

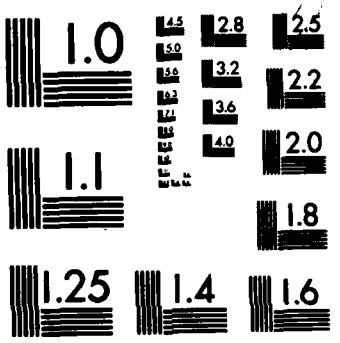
Science Applications, Inc
ATTN: P. Setty
ATTN: W. Zimmerman
ATTN: W. Layson
5 cy ATTN: Security Ofc

Science Applications, Inc
ATTN: R. McClean

Science Applications, Inc
ATTN: H. Dolatshahi
ATTN: R. Jarka

AD-A131 273 AFWL BENCHMARK COMPUTER PROGRAM(U) BOEING AEROSPACE CO 2/2
SEATTLE WA B J HENDERSON 01 MAY 82 DNA-5852F
DNA001-80-C-0056

UNCLASSIFIED F/G 9/2 NL



DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Science Applications, Inc
ATTN: C. Bates

Scientific Rsch Assoc, Inc
ATTN: J. Kredozsky

SRI International
ATTN: P. Wong
ATTN: B. Lew
ATTN: G. Abrahamson
ATTN: P. Dolan
9 cy ATTN: Library

SRI International
2 cy ATTN: Doc Con

Sylvania Systems Grp
2 cy ATTN: Library

Strategic Systems Div
ATTN: A. Novenski

System Planning & Analysis, Inc
ATTN: Contracts Ofc
ATTN: P. Vadola

Tetra Tech, Inc
ATTN: Contracts Ofc

Tetra Tech, Inc
ATTN: J. Preston

Toyon Rsch Corp
ATTN: J. Garbarino

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

TRW Electronics & Defense Sector
ATTN: G. Teraoka
ATTN: D. Jortner
ATTN: F. Fendell
2 cy ATTN: Tech Info Ctr

TRW Electronics & Defense Sector
ATTN: F. Sterk
ATTN: F. Straw

United Technologies Rsch Ctr
ATTN: H. Michels

Visidyne, Inc
ATTN: H. Smith

Weidlinger Assoc, Consulting Engrg
ATTN: J. Wright
3 cy ATTN: M. Baron

Weidlinger Assoc, Consulting Engrg
ATTN: K. Stultz

Weidlinger Associates
ATTN: J. Isenberg
ATTN: E. Richardson

Weidlinger Associates, Inc
ATTN: E. Raney

Westinghouse Corp
ATTN: J. Kasdorf